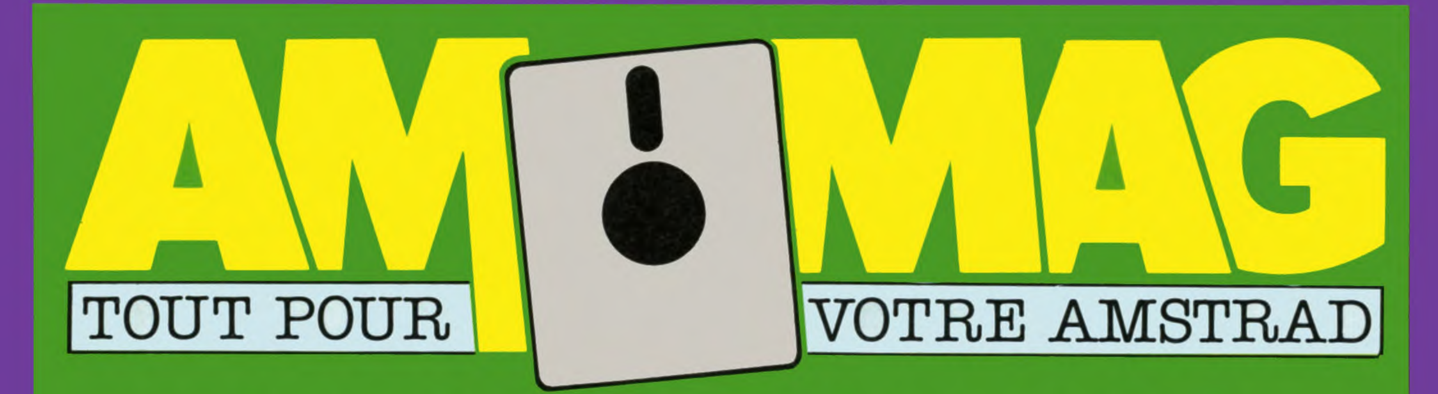


SPECIAL LISTINGS

HORS-SERIE N° 10

60 pages ; « jeux et utilitaires »





FEVRIER 1989. 25 F

HORS-SERIE N° 10

Février 1989

SOMMAIRE



Ce "Hors-série" d'AM-MAG est édité par **Laser Presse SA, 5-7, rue de l'Amiral Courbet 94160 SAINT-MANDE.**
Directeur de la publication : Jean Kaminsky.
REDACTION. Directeur technique de la rédaction : Xavier Frigara.
Directeur de la rédaction : Jacques Eltabet. **Secrétaire de rédaction :** Gaëlle Pillot.
Chef de la rubrique programmation : Jean-Claude Paulin.
Ont collaboré à ce numéro : Fabien Gaudin, Luc et Hervé Guillaume, Hervé Inisan, Claude Le Moullec, Frédéric Carbonero, Christophe Leclercq, Sébastien Royer.
FABRICATION. Directeur de la fabrication : Jean-Jacques Galmiche. **Maquettistes :** Thierry Martinez, Laurence Floquet, Michel Lhopitault. **Montage et photogravure :** Jean-Baptiste Ballériaud, Michel Lhopitault.
ADMINISTRATION. Diffusion : Bertrand Desroche. **Abonnements :** Martine Lapierre au (1) 43.98.01.71. **Comptabilité :** Sylvie Kaminsky.
REGIE PUBLICITAIRE : NEO-MEDIA, 5-7, rue de l'Amiral Courbet 94160 SAINT-MANDE. Tél. : (1) 43.98.22.22.
Directeur commercial : Philippe Fanovard. **Chef de publicité :** Thierry Cagnion.
Assistante de publicité : Mick Deret.
Commission paritaire : en cours. **Dépôt légal :** 1er trimestre 1989.
Photocomposition : Composcopie. **Impression :** RBI, Tima Roto.

UTILITAIRE

Ce programme figure sur la disquette : HORS-SERIE N° 10

VÉRIFICATEURS V.2

Le saviez-vous ? La présence dans nos listings de numéros entre crochets, assure la détection d'éventuelles erreurs lors de la saisie d'un programme Basic.
Tapez minutieusement et sauvegardez le listing vérificateur relatif à votre machine (464, 664 ou 6128), conservez-le précieusement et lancez-le avant toute frappe ou vérification de listing ; les indications nécessaires sont incluses. Chaque somme affichée devra rigoureusement correspondre à celle figurant dans nos colonnes.
Nota : les deux traits verticaux présents en lignes 80 et 90, correspondent à une barre verticale sur un clavier Qwerty, ou à un « ù » sur clavier Azerty. Ces caractères —CHR$ (124) — s'obtiennent par l'appui simultané sur SHIFT/ traits verticaux.

Avec l'aimable autorisation de CPC Schneider International.

```
10 REM Verificateur 464
20 MEMORY &A4FF
30 FOR a%=&A500 TO &A607
40 READ byte$
50 POKE a%,VAL("&"+byte$)
60 NEXT
70 CLS:LOCATE 5,1:PRINT CHR$(24)
" Verificateur V.2 464 installe
"CHR$(24):PRINT:PRINT"Chaque val
idation affiche desormais une so
mme de controle.":PRINT
80 PRINT"|CHECK,2 : liste le pro
gramme avec les            somm
es de controle.":PRINT:PRINT"|CH
ECK,8 : imprime le programme ave
c les          sommes de contro
le.":PRINT:PRINT"    "CHR$(24)"
Taper ou charger un programme. "
CHR$(24)
90 CALL &A500:|ON
100 NEW
110 DATA 21,09,a5,01,0d,a5,c3,d1
120 DATA bc,00,00,00,00,18,a5,c3
130 DATA 2a,a5,c3,2f,a5,c3,43,a5
140 DATA 4f,ce,4f,46,c6,43,48,45
150 DATA 43,cb,00,00,cf,98,aa,c3
160 DATA a8,a5,21,27,a5,18,03,21
170 DATA 24,a5,28,06,cd,00,b9,c3
180 DATA 06,dd,11,3a,bd,01,03,00
190 DATA ed,b0,c9,4f,cd,00,b9,0d
200 DATA 28,08,0d,20,ea,dd,7e,02
210 DATA 18,04,7b,11,01,00,cd,a2
220 DATA c1,cd,a3,e7,e5,4e,23,46
230 DATA 23,5e,23,56,e1,78,b1,c8
240 DATA cd,3c,c4,e5,09,e3,cd,63
250 DATA e1,21,a4,ac,cd,7a,a5,e1
260 DATA 18,e2,e5,cd,ba,a5,e3,cd
270 DATA 98,a5,cd,96,f2,e3,cd,f6
280 DATA a5,cd,4e,c3,e1,7e,a7,c8
290 DATA cd,98,a5,cd,4e,c3,18,f5
300 DATA 3a,24,ac,d6,08,47,7e,a7
310 DATA c8,cd,45,e1,23,10,f7,c9
320 DATA cd,24,a5,f5,c5,d5,e5,cd
330 DATA ba,a5,cd,f6,a5,e1,d1,c1
340 DATA f1,c9,eb,1b,af,47,67,6f
350 DATA 2f,32,23,a5,13,1a,d6,30
360 DATA 38,04,fe,0a,38,f6,1a,13
370 DATA a7,c8,4f,3a,23,a5,a1,fe
380 DATA 20,28,f3,79,fe,22,20,07
390 DATA 3a,23,a5,2f,32,23,a5,3a
400 DATA 23,a5,a7,79,c4,8a,ff,4f
410 DATA ad,07,6f,09,18,d8,3e,20
420 DATA cd,5c,c3,3e,5b,cd,5c,c3
430 DATA cd,79,ee,3e,5d,c3,5c,c3
```

```
10 REM Verificateur 664
20 MEMORY &A4FF
30 FOR a%=&A500 TO &A607
40 READ byte$
50 POKE a%,VAL("&"+byte$)
60 NEXT
70 CLS:LOCATE 5,1:PRINT CHR$(24)
" Verificateur V.2 664 installe
"CHR$(24):PRINT:PRINT"Chaque val
idation affiche desormais une so
mme de controle.":PRINT
80 PRINT"|CHECK,2 : liste le pro
gramme avec les            somm
es de controle.":PRINT:PRINT"|CH
ECK,8 : imprime le programme ave
c les          sommes de contro
le.":PRINT:PRINT"    "CHR$(24)"
Taper ou charger un programme. "
CHR$(24)
90 CALL &A500:|ON
100 NEW
110 DATA 21,09,a5,01,0d,a5,c3,d1
120 DATA bc,00,00,00,00,18,a5,c3
130 DATA 2a,a5,c3,2f,a5,c3,43,a5
140 DATA 4f,ce,4f,46,c6,43,48,45
150 DATA 43,cb,00,00,cf,02,ac,c3
160 DATA a8,a5,21,27,a5,18,03,21
170 DATA 24,a5,28,06,cd,00,b9,c3
180 DATA 4c,cb,11,5b,bd,01,03,00
190 DATA ed,b0,c9,4f,cd,00,b9,0d
200 DATA 28,08,0d,20,ea,dd,7e,02
210 DATA 18,04,7b,11,01,00,cd,a9
220 DATA c1,cd,69,e8,e5,4e,23,46
230 DATA 23,5e,23,56,e1,78,b1,c8
240 DATA cd,75,c4,e5,09,e3,cd,59
250 DATA e2,21,8a,ac,cd,7a,a5,e1
260 DATA 18,e2,e5,cd,ba,a5,e3,cd
270 DATA 98,a5,cd,58,f3,e3,cd,f6
280 DATA a5,cd,9b,c3,e1,7e,a7,c8
290 DATA cd,98,a5,cd,9b,c3,18,f5
300 DATA 3a,09,ac,d6,08,47,7e,a7
310 DATA c8,cd,22,e2,23,10,f7,c9
320 DATA cd,24,a5,f5,c5,d5,e5,cd
330 DATA ba,a5,cd,f6,a5,e1,d1,c1
340 DATA f1,c9,eb,1b,af,47,67,6f
350 DATA 2f,32,23,a5,13,1a,d6,30
360 DATA 38,04,fe,0a,38,f6,1a,13
370 DATA a7,c8,4f,3a,23,a5,a1,fe
380 DATA 20,28,f3,79,fe,22,20,07
390 DATA 3a,23,a5,2f,32,23,a5,3a
400 DATA 23,a5,a7,79,c4,ab,ff,4f
410 DATA ad,07,6f,09,18,d8,3e,20
420 DATA cd,a3,c3,3e,5b,cd,a3,c3
430 DATA cd,49,ef,3e,5d,c3,a3,c3
```

```
10 REM Verificateur 6128
20 MEMORY &A4FF
30 FOR a%=&A500 TO &A607
40 READ byte$
50 POKE a%,VAL("&"+byte$)
60 NEXT
70 CLS:LOCATE 4,1:PRINT CHR$(24)
" Verificateur V.2 6128 installe
 "CHR$(24):PRINT:PRINT"Chaque va
lidation affiche desormais une s
omme de controle.":PRINT
80 PRINT"|CHECK,2 : liste le pro
gramme avec les            somm
es de controle.":PRINT:PRINT"|CH
ECK,8 : imprime le programme ave
c les          sommes de contro
le.":PRINT:PRINT"    "CHR$(24)" T
aper ou charger un programme. "C
HR$(24)
90 CALL &A500:|ON
100 NEW
110 DATA 21,09,a5,01,0d,a5,c3,d1
120 DATA bc,00,00,00,00,18,a5,c3
130 DATA 2a,a5,c3,2f,a5,c3,43,a5
140 DATA 4f,ce,4f,46,c6,43,48,45
150 DATA 43,cb,00,00,cf,02,ac,c3
160 DATA a8,a5,21,27,a5,18,03,21
170 DATA 24,a5,28,06,cd,00,b9,c3
180 DATA 49,cb,11,5e,bd,01,03,00
190 DATA ed,b0,c9,4f,cd,00,b9,0d
200 DATA 28,08,0d,20,ea,dd,7e,02
210 DATA 18,04,7b,11,01,00,cd,a6
220 DATA c1,cd,64,e8,e5,4e,23,46
230 DATA 23,5e,23,56,e1,78,b1,c8
240 DATA cd,72,c4,e5,09,e3,cd,54
250 DATA e2,21,8a,ac,cd,7a,a5,e1
260 DATA 18,e2,e5,cd,ba,a5,e3,cd
270 DATA 98,a5,cd,53,f3,e3,cd,f6
280 DATA a5,cd,98,c3,e1,7e,a7,c8
290 DATA cd,98,a5,cd,98,c3,18,f5
300 DATA 3a,09,ac,d6,08,47,7e,a7
310 DATA c8,cd,1d,e2,23,10,f7,c9
320 DATA cd,24,a5,f5,c5,d5,e5,cd
330 DATA ba,a5,cd,f6,a5,e1,d1,c1
340 DATA f1,c9,eb,1b,af,47,67,6f
350 DATA 2f,32,23,a5,13,1a,d6,30
360 DATA 38,04,fe,0a,38,f6,1a,13
370 DATA a7,c8,4f,3a,23,a5,a1,fe
380 DATA 20,28,f3,79,fe,22,20,07
390 DATA 3a,23,a5,2f,32,23,a5,3a
400 DATA 23,a5,a7,79,c4,ab,ff,4f
410 DATA ad,07,6f,09,18,d8,3e,20
420 DATA cd,a0,c3,3e,5b,cd,a0,c3
430 DATA cd,44,ef,3e,5d,c3,a0,c3
```

UTILITAIRE



AMSAISIE

Nous vous félicitons d'être de plus en plus nombreux à nous communiquer des programmes réalisés partiellement ou totalement en langage machine. Les avantages de ce type de programmation ne sont plus à démontrer, ainsi que l'exceptionnelle qualité de la plupart des créations.

Malheureusement, la saisie par le lecteur d'une pléthore de codes machine n'est pas une sinécure. *Amsaisie* a donc été créé dans le but louable de vous faire profiter de quelques petites merveilles, en vous évitant un maximun de contraintes.

Il doit être conservé précieusement et utilisé chaque fois que vous découvrirez dnas nos colonnes, un listing au format approprié.

Exemple :

Adresse	Codes	Somme de contrôle
8CF0	E5 DD E1 3A F6 2F 3C 11	CB

Mode d'emploi

Après lancement, spécifiez en hexadécimal (sans le préfixe « & ») l'adresse de début d'implantation du langage machine. Celle-ci s'affiche, suivie de « : » et d'un curseur clignotant. Entrez la série de huit codes sans vous préoccuper des espaces et sans valider par RETURN (cela est réalisé automatiquement).

En fin de ligne et à l'affichage de « : », entrez la somme de contrôle correspondant à la ligne saisie. En l'absence d'erreur, l'adresse suivante s'affiche, etc. Dans le cas contraire, un signal sonore et le message « ERREUR! » vous signalent une bévue. Pas de panique, puisque l'adresse de la ligne litigieuse (qu'il suffira de retaper) se réaffiche. De plus, sachez qu'en cours de saisie, la touche DEL est opérationnelle.

Maintenant, comment sauvegarder sur disquette (ou cassette) le langage machine aussi facilement entré en mémoire ? Tout simplement par la commande « S » qui vous réclamera préalablement le nom du programme à sauver. Toutefois, deux solutions s'offrent à vous.

— Vous êtes fou et venez de saisir en une seule fois, la « totalité » des codes héxadécimaux (très nombreux dans la plupart des cas). Pas de problème, après l'entrée de la dernière somme de contrôle et l'affichage de l'adresse suivante, appuyez sur « S », précisez le nom du fichier et validez par RETURN (ou ENTER).

— Vous êtes raisonnable et désirez morceler votre saisie. Au moment de stopper momentanément votre ouvrage pour le poursuivre ultérieurement, appuyez sur « S » après l'affichage de l'adresse suivante et attribuez un numéro d'ordre à votre nom de fichier (exemple : PENDU1). À la fin de la sauvegarde, l'adresse suivante déjà citée se réaffiche ; notez-la. Elle sera « l'adresse de début » qu'il conviendra de spécifier lors de la reprise de votre travail (PENDU2).

Créez de la sorte, une suite de fichiers binaires (PENDU1, PENDU2, PENDU3, etc.). Finalement, chargez à la suite tous ces fichiers après un MEMORY adéquat (inférieur d'une unité à l'adresse d'implantation) et effectuez une sauvegarde totale et définitive par la commande de type :
SAVE "nom de fichier", b, Adresse de début, longueur
Le nom du fichier et la valeur des paramètres sont toujours précisés dans le mode d'emploi des programmes publiés. Prenons l'exemple d'un programme binaire appelé PENDU, d'adresse de début &A000 et morcelé en trois fichiers. Pour les réunir en un seul d'après la longueur totale &BFF indiquée dans le mode d'emploi, il faudra lancer le programme suivant (ou rentrer les commandes en direct) :
10 MEMORY &A000-1
20 LOAD"PENDU1.bin" : LOAD"PENDU2.bin": LOAD"PENDU3.bin"
30 SAVE"PENDU",b,&A000, &BFF
Enfantin !

```
10 ' * AMSAISIE par Denis JARRIL *  [1660]
20 MEMORY &2000:DIM O$(18):MODE 1:B [17671]
ORDER 0:INK 0,0:INK 1,13:CLS:PRINT:
PRINT" I pour changer l'adresse cou
rante":PRINT" S pour sauver les don
nees":PRINT" Tapez les caracteres s
ans espace ni     return (tout se f
ait automatiquement).":PRINT
30 PRINT:INPUT" ADRESSE DE DEPART : [4110]
 ",A$:D$=A$:IF A$="" THEN 30
40 A=VAL("&"+A$)                    [1273]
50 I=0:A$=HEX$(A,4):PRINT:PRINT A$; [3200]
":";:C=VAL("&"+LEFT$(A$,2))+VAL("&"
+RIGHT$(A$,2))
60 T$="":WHILE T$="":CALL &BB8A:T$= [3454]
INKEY$:CALL &BB8D:WEND:T$=UPPER$(T$
)
70 IF T$="I" THEN CLS:RUN           [1132]
80 IF T$<>"S" THEN 110 ELSE D=VAL(" [3318]
&"+D$):IF D>0 AND A<0 THEN A=A+6553
6
90 PRINT:PRINT:INPUT"  NOM : ",N$:I [3394]
F N$<>"" THEN SAVE N$,B,D,A-D+1
100 GOTO 50                          [384]
110 IF T$<>CHR$(127) THEN 130 ELSE  [7405]
IF I=0 THEN 60 ELSE I=I-1:PRINT CHR
$(8);" ";CHR$(8);:IF I/2<>ROUND(I/2
) THEN PRINT CHR$(8);" ";CHR$(8);
120 GOTO 60                          [390]
130 IF T$<"0" OR T$>"F" THEN SOUND  [2693]
7,150,20:GOTO 60
140 IF T$<"A" AND T$>"9" THEN SOUND [2632]
 7,150,20:GOTO 60
150 PRINT T$;:IF I=15 THEN PRINT":" [3278]
; ELSE IF I/2<>ROUND(I/2) THEN PRIN
T" ";
160 O$(I)=T$                         [592]
170 I=I+1:IF I<18 THEN 60           [1324]
180 FOR I=0 TO 15 STEP 2:X=VAL("&"+ [4450]
O$(I)+O$(I+1)):POKE A,X:A=A+1:C=C+X
:NEXT:C=C AND &FF
190 IF C=VAL("&"+O$(I)+O$(I+1)) THE [2426]
N 50
200 SOUND 7,50,10:SOUND 7,500,10:PR [3493]
INT"ERREUR!";:GOTO 40
```





JEASON

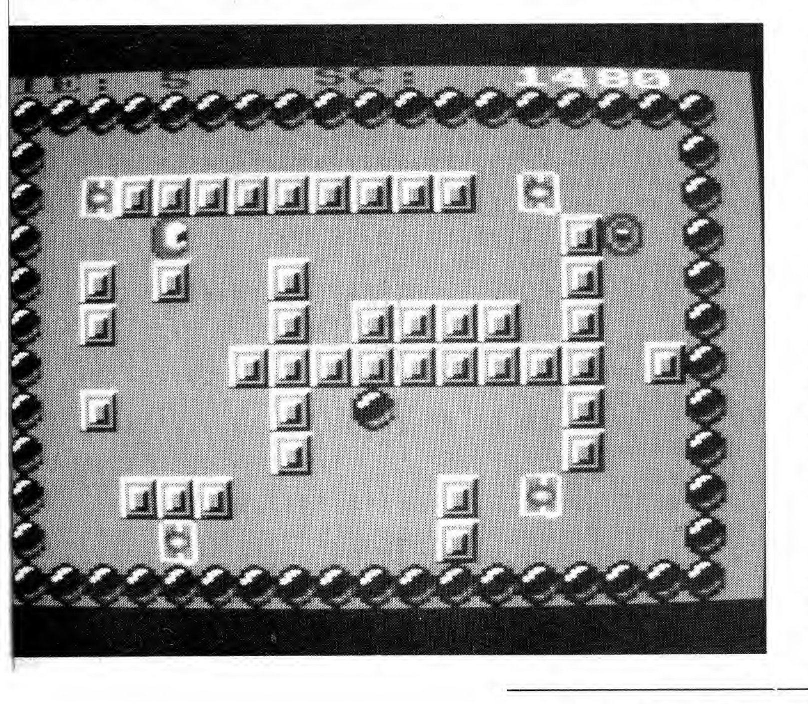
Comment détruire à l'aide de briques de feu, des boules sacrées réparties dans les vingt-cinq salles d'un temple gigantesque particulièrement mal fréquenté ? Par la malice et la perspicacité qui caractérisent notre ami Jeason...

Sauvegarde

Sauvez tout d'abord sous un nom de votre choix (JEASON) le premier listing Basic. Entrez ensuite par *Amsaisie* (reportez-vous à son mode d'emploi) le court listing de codes hexadécimaux. Spécifiez 7000 comme adresse de début et sauvegardez le langage machine par l'option « S » sous le nom « JEASON1 ». Sauvegardez ensuite le troisième listing Basic (programme principal) sous le nom « JEASON2 ». Les règles sont incluses.

Nota : le caractère de contrôle présent dans les lignes 2480, 2500 et 2520, s'obtient par appui simultané sur CTRL/O.

Sébastien Royer



```
10 '***************                    [528]
20 '****          ****                 [278]
30 '**** JEASON ****                   [684]
40 '****          ****                 [278]
50 '***************                    [528]
60 '*** S. ROYER ***                   [883]
70 '***************                    [528]
80 BORDER 0:DIM C(15):MODE 0:FOR I= [3747]
0 TO 15:READ A:INK I,0:C(I)=A:NEXT
90 DATA 0,1,2,14,20,19,18,9,3,6,15, [1712]
24,25,26,8,7
100 PRINT CHR$(23);CHR$(3)          [1990]
110 LOCATE 7,3:PEN 14:PRINT"JEASON" [2394]
120 LOCATE 9,6:PEN 15:PRINT"(C)"    [1459]
130 LOCATE 5,9:PEN 14:PRINT"R.O.S S [2801]
OFT"
140 LOCATE 9,12:PEN 15:PRINT"BY"    [1366]
150 LOCATE 3,15:PEN 14:PRINT"SEBAST [3561]
IEN ROYER"
160 C=1                             [351]
170 FOR I=0 TO 400 STEP 4           [1513]
180 PLOT 0,I,C MOD 13+1             [444]
190 DRAWR 576,0                     [744]
200 PLOT 0,I+2,C MOD 13+1           [616]
210 DRAWR 576,0                     [744]
220 C=C+1                           [701]
230 NEXT                            [350]
240 INK 14,C(14):INK 15,C(15)       [1939]
250 FOR O=1 TO 80                   [941]
260 A=C(1)                          [806]
270 FOR I=1 TO 12                   [321]
280 C(I)=C(I+1)                     [389]
290 INK I,C(I)                      [670]
300 NEXT                            [350]
310 CALL &BD19                      [352]
320 INK 13,A                        [286]
330 C(13)=A                         [462]
340 NEXT                            [350]
350 MEMORY &7000-1                  [483]
360 LOAD"!JEASON1"                  [1572]
370 RUN"!JEASON2"                   [519]
```

```
7000:00 04 08 00 00 0C 0C 00:94
7008:04 5C AC 08 04 FC 0C 08:A0
7010:0C AC 0C 0C 5C AC 0C 0C:70
7018:5C 0C 0C 4C 0C 0C 0C 9C:08
7020:0C 0C 0C 3C 0C 0C 4C BE:12
7028:0C 0C 9C AE 0C 4C 7D 2C:FB
7030:04 7D BE 08 04 9C 6C 08:FB
7038:00 4C 8C 00 00 04 08 00:8C
7040:FC FC FC FC FC FC FC AC:40
7048:B8 30 30 24 B8 30 30 24:30
7050:B8 FC FC 24 B8 FC AC 24:18
7058:B8 B8 24 24 B8 B8 24 24:38
7060:B8 B8 24 24 B8 B8 24 24:40
7068:B8 B8 24 24 B8 AC 0C 24:24
7070:B8 0C 0C 24 B8 30 30 24:10
7078:AC 0C 0C 0C 0C 0C 0C 0C:E8
7080:54 FC FC A8 B8 30 B8 74:F8
7088:A8 10 A8 54 A8 10 A8 54:60
7090:A8 34 38 54 A8 3C 3C 54:DC
7098:BC 7D BE 7C BC AB 57 7C:B5
70A0:BC AB 57 7C BC 7D BE 7C:BD
70A8:A8 3C 3C 54 A8 34 38 54:F4
70B0:A8 10 A8 54 A8 10 A8 54:88
70B8:FC FC FC FC 10 30 74 A8:74
70C0:00 3C 3C 00 14 3C FF 28:1F
70C8:14 7D 56 28 3C AB FC BC:E6
70D0:3C AB FC BC 3C AB FC 1C:DE
70D8:3C AB AC 1C 3C AB AC 1C:A6
70E0:3C AB AC 1C 3C AB 06 16:02
70E8:3C FF 03 BE 3C 7D FF 3C:48
70F0:3C 3C 3C 94 14 68 C0 28:0C
70F8:14 3C 3C 28 00 3C 3C 00:94
7100:00 3C 3C 00 14 FF 3C 28:60
7108:14 A9 BE 28 7C FC 57 3C:27
7110:7C FC 57 3C 2C FC 57 3C:47
7118:2C 5C 57 3C 2C 5C 57 3C:BF
7120:2C 5C 57 3C 29 09 57 3C:71
7128:7D 03 FF 3C 3C FF BE 3C:89
7130:68 3C 3C 3C 14 C0 94 28:4D
7138:14 3C 3C 28 00 3C 3C 00:D5
7140:00 3C 3C 00 54 3C 7D 02:38
7148:54 57 AB A8 FC 57 FE FC:04
7150:AC 57 FE 5C AC 5D AE 5C:31
7158:AC 5D AE 5C 06 5D AE 5C:49
7160:AB 5D AE FD 7D 57 FE BE:14
7168:3C BE 7D 3C 3C 3C 3C 3C:7C
```

```
7170:68 3C 3C 94 14 C0 C0 28:11
7178:14 3C 3C 28 00 3C 3C 00:15
7180:00 3C 3C 00 14 3C 3C 28:1D
7188:14 3C 3C 28 3C 3C 3C 3C:9D
7190:3C 3C 3C 3C 3C 3C 3C 3C:E1
7198:3C 3C 3C 3C 3C 3C 3C 3C:E9
71A0:3C 3C 3C 3C 3C 3C 3C 3C:F1
71A8:3C 3C 3C 3C 3C 3C 3C 3C:F9
71B0:3C 3C 3C 3C 14 3C 3C 28:C5
71B8:14 3C 3C 28 00 3C 3C 00:55
71C0:00 C3 C3 00 41 C3 C3 82:00
71C8:41 87 4B 82 C3 1E 2D C3:9F
71D0:C3 1E 2D C3 87 3C 3C 4B:5C
71D8:87 7C BC 4B 87 2C 1C 4B:6D
71E0:87 2C 1C 4B 87 2C 1C 4B:85
71E8:87 3C 3C 4B C3 1E 2D C3:74
71F0:C3 1E 2D C3 41 87 4B 82:C7
71F8:41 C3 C3 82 00 C3 C3 00:38
7200:00 CF CF 00 45 33 33 8A:45
7208:45 67 9B 8A 9B 9B 67 67:4F
7210:9B 9B 67 67 9B 9B 67 67:8A
7218:9B 9B 67 67 9B 9B 67 67:92
7220:9B 9B 67 67 9B 9B 67 67:9A
7228:9B 9B 67 67 9B 9B 67 67:A2
7230:9B 9B 67 67 45 67 9B 8A:77
7238:45 33 33 8A 00 CF CF 00:7D
7240:00 00 00 00 00 00 00 00:B2
```

```
10 DEFINT A-Z:ENV 1,15,1,1,15,-0.5, [3681]
3:ENT -1,3,1,1,9,-1,1:ENV 2,15,-1,1
:FOR I=0 TO 15
20 READ A:INK I,A:NEXT                 [1483]
30 DATA 2,1,0,3,13,0,6,26,25,4,5,18   [1800]
,9,0,0,15
40 BORDER 0:DIM X(17,11),SP(10)        [1615]
50 DEF FNPO(X,Y)=&C0F8+(Y-1)*160+(X   [3340]
-1)*4:DEF FNPOT(X,Y)=&C0F8+(Y-1)*80
+(X-1)*4
60 MODE 0:FOR I=0 TO 9:SP(I+1)=&700   [2366]
0+I*64:NEXT
70 IF PEEK(&9000)=221 THEN GOTO 240   [1972]
0
80 FOR I=&9000 TO &9063                [1598]
90 READ A$:POKE I,VAL("&"+A$):NEXT     [614]
100 DATA DD,5E,00,DD,56,01,DD,6E       [586]
110 DATA 02,DD,66,03,06,05,C5,D5       [1639]
120 DATA 11,C0,70,CD,49,90,D1,23       [1342]
130 DATA 23,23,23,CD,49,90,2B,2B       [1140]
140 DATA 2B,C1,C5,78,FE,01,28,07       [1230]
150 DATA 2B,CD,33,90,23,00,00,C1       [1400]
160 DATA 10,DC,C9                      [447]
170 DATA E5,06,10,C5                   [796]
180 DATA E5,06,01,3E,00,77,23,10       [975]
190 DATA FA,E1,CD,26,BC,C1,10,EF       [1591]
200 DATA E1,C9,CD,19,BD,D5,E5,06       [825]
210 DATA 10,C5,E5,06,04,1A,77,23       [1245]
220 DATA 13,10,FA,E1,CD,26,BC,C1,10    [1089]
230 DATA EF,E1,D1,C9                   [134]
240 FOR I=&A000 TO &A00F               [626]
250 READ A$:POKE I,VAL("&"+A$):NEXT    [614]
260 DATA DD,5E,00,DD,56,01,DD,6E       [586]
270 DATA 02,DD,66,03,CD,49,90,C9       [1324]
280 FOR I=&9100 TO &913A               [1100]
290 READ A$:POKE I,VAL("&"+A$):NEXT    [614]
300 DATA DD,5E,00,DD,56,01,DD,6E       [586]
310 DATA 02,DD,66,03,06,05,C5,CD       [1822]
320 DATA 49,90,23,23,23,23,D5,11       [1549]
330 DATA 00,71,CD,49,90,D1,C1,C5       [1535]
340 DATA 78,FE,01,28,0D,23,23,23       [1248]
350 DATA 00,00,CD,33,90,06,03,2B       [1186]
360 DATA 10,FD,06,05,2B,10,FD,C1       [1167]
370 DATA 10,D4,C9                      [642]
380 FOR I=&9200 TO &9269               [1163]
390 READ A$:POKE I,VAL("&"+A$):NEXT    [614]
400 DATA DD,5E,00,DD,56,01,DD,6E       [586]
410 DATA 02,DD,66,03,06,05,C5,CD       [1822]
420 DATA 49,90,06,04,CD,3C,92,D5       [1210]
430 DATA 11,80,71,CD,49,90,D1,C1       [1391]
440 DATA C5,78,FE,01,28,0D,06,03       [1184]
450 DATA CD,3C,92,CD,54,92,06,03       [1140]
460 DATA CD,48,92,06,05,CD,48,92       [849]
470 DATA C1,10,D3,C9                   [753]
480 DATA C5,06,04,CD,26,BC,10,FB,C1    [1808]
,10,F5,C9
490 DATA C5,06,04,CD,29,BC,10,FB,C1    [2382]
,10,F5,C9
500 DATA E5,06,04,C5                   [646]
510 DATA E5,06,04,3E,00,77,23,10       [845]
520 DATA FA,E1,CD,26,BC,C1,10,EF       [1591]
530 DATA E1,C9                         [486]
540 FOR I=&9300 TO &9331               [1658]
550 READ A$:POKE I,VAL("&"+A$):NEXT    [614]
560 DATA DD,5E,00,DD,56,01,DD,6E       [586]
570 DATA 02,DD,66,03,06,05,C5,CD       [1822]
580 DATA 49,90,06,04,CD,48,92,D5       [1272]
590 DATA 11,40,71,CD,49,90,D1,C1       [904]
600 DATA C5,78,FE,01,28,03,CD,54       [1055]
610 DATA 92,06,05,CD,3C,92,C1,10       [1514]
620 DATA DD,C9                         [403]
630 FOR I=&9400 TO &9420               [1261]
640 READ A$:POKE I,VAL("&"+A$):NEXT    [614]
650 DATA DD,5E,00,DD,56,01,DD,6E       [586]
660 DATA 02,DD,66,03,06,05,C5,CD       [1822]
670 DATA 49,90,C1,C5,78,FE,01,28       [1828]
680 DATA 03,CD,33,90,23,C1,10,EE       [1372]
690 DATA C9                            [276]
700 FOR I=&9500 TO &9526               [1188]
710 READ A$:POKE I,VAL("&"+A$):NEXT    [614]
720 DATA DD,5E,00,DD,56,01,DD,6E       [586]
730 DATA 02,DD,66,03,06,05,C5,CD       [1822]
740 DATA 49,90,C1,C5,78,FE,01,28       [1828]
750 DATA 09,23,23,23,CD,33,90,2B       [956]
760 DATA 2B,2B,2B,C1,10,E8,C9          [1134]
770 FOR I=&9600 TO &9629               [1163]
780 READ A$:POKE I,VAL("&"+A$):NEXT    [614]
790 DATA DD,5E,00,DD,56,01,DD,6E       [586]
800 DATA 02,DD,66,03,06,05,C5,CD       [1822]
810 DATA 49,90,06,03,CD,3C,92,C1,C5    [1721]
,78,FE,01,28
820 DATA 03,CD,54,92,06,04,CD,48       [1681]
830 DATA 92,C1,10,E5,C9                [70]
840 FOR I=&9700 TO &9724               [1194]
850 READ A$:POKE I,VAL("&"+A$):NEXT    [614]
860 DATA DD,5E,00,DD,56,01,DD,6E       [586]
870 DATA 02,DD,66,03,06,05,C5,CD       [1822]
880 DATA 49,90,C1,C5,78,FE,01,28       [1828]
890 DATA 03,CD,54,92,06,01,CD,3C       [1481]
900 DATA 92,C1,10,EA,C9                [331]
910 GOTO 2400                          [347]
920 ' *****                            [174]
930 ' *JEU*                            [230]
940 ' *****                            [174]
950 MODE 0:VIE=5:SC=0:TA=1             [1971]
960 WINDOW#1,3,18,4,23                 [1052]
970 LOCATE 1,1:PEN 6:PRINT"VIE:";      [1197]
980 PEN 9:PRINT VIE                    [1579]
990 LOCATE 10,1:PEN 1:PRINT"SC:";      [1905]
1000 GOSUB 1490                        [825]
1010 FOR I=1 TO 18                     [339]
1020 CALL &A000,FNPO(I,1)-164,SP(1)    [1837]
1030 CALL &A000,FNPO(I,11)-4,SP(1)     [1430]
1040 NEXT                              [350]
1050 FOR I=1 TO 11                     [314]
1060 CALL &A000,FNPO(1,I)-4,SP(1)      [1331]
1070 CALL &A000,FNPO(17,I),SP(1)       [1279]
```

```
1080 NEXT                                   [350]
1090 GOTO 1770                              [361]
1100 END                                    [110]
1110 ' *********************                [673]
1120 ' * ROUTINE PRINCIPALE *               [822]
1130 ' *********************                [673]
1140 MON=0                                  [482]
1150 EVERY (25-MSTR)*500,0 GOSUB 23 [966]
10
1160 X5=X:Y5=Y                              [904]
1170 DI:IF INKEY(GA)=0 AND X>1 THEN [12293]
 IF X(X-1,Y)=0 THEN CALL &9500,FNPO
(X,Y),SP(5):X=X-1:GOTO 1210 ELSE IF
 X>2 THEN IF X(X-2,Y)=0 THEN X(X-2,
Y)=X(X-1,Y):X(X-1,Y)=0:X=X-1:CALL &
9100,FNPO(X,Y),SP(X(X-1,Y)):IF X(X-
1,Y)=1 THEN XB=X-1:GOTO 1210
1180 IF INKEY(DA)=0 AND X<16 THEN I [13662]
F X(X+1,Y)=0 THEN CALL &9400,FNPO(X
,Y),SP(4):X=X+1:GOTO 1210 ELSE IF X
<15 THEN IF X(X+2,Y)=0 THEN X(X+2,Y
)=X(X+1,Y):X(X+1,Y)=0:CALL &9000,FN
PO(X,Y),SP(X(X+2,Y)):X=X+1:IF X(X+1
,Y)=1 THEN XB=X+1:GOTO 1210
1190 IF INKEY(HT)=0 AND Y>1 THEN IF [14265]
 X(X,Y-1)=0 THEN CALL &9600,FNPO(X,
Y),SP(7):Y=Y-1:GOTO 1210 ELSE IF Y>
2 THEN IF X(X,Y-2)=0 THEN X(X,Y-2)=
X(X,Y-1):X(X,Y-1)=0:Y=Y-1:CALL &920
0,FNPO(X,Y),SP(X(X,Y-1)):IF X(X,Y-1
)=1 THEN YB=Y-1:GOTO 1210
1200 IF INKEY(BA)=0 AND Y<10 THEN I [12361]
F X(X,Y+1)=0 THEN CALL &9700,FNPO(X
,Y),SP(6):Y=Y+1 ELSE IF Y<9 THEN IF
 X(X,Y+2)=0 THEN X(X,Y+2)=X(X,Y+1):
X(X,Y+1)=0:Y=Y+1:CALL &9300,FNPO(X,
Y),SP(X(X,Y+1)):IF X(X,Y+1)=1 THEN
YB=Y+1
1210 EI                                     [86]
1220 IF X<>X5 OR Y<>Y5 THEN SC=SC+5 [2317]
:GOSUB 1490
1230 IF X=X1 AND Y=Y1 THEN FIN=1        [1266]
1240 IF INKEY(AB)=0 THEN FIN=1          [2062]
1250 IF X(XB-1,YB)=3 AND X(XB+1,YB) [9262]
=3 AND X(XB,YB-1)=3 AND X(XB,YB+1)=
3 THEN DI:TA=TA+1:FOR I=200 TO 0 ST
EP -20:SOUND 1,I,4,15:NEXT:GOTO 177
0
1260 IF X(X1,Y1)<>0 THEN DI:X1=XP:Y [6730]
1=YP:SC=SC+500:X(X1,Y1)=0:CALL &A00
0,FNPO(X1,Y1),SP(10):CALL &A000,FNP
O(X1,Y1),SP(8):MON=0:GOSUB 1490:EI
1270 MON=MON+1:IF MON=MSTR THEN MON [2144]
=0:GOSUB 1300
1280 IF FIN=1 THEN MON=0:FIN=0:GOTO [1750]
 1410
1290 GOTO 1160                              [307]
1300 ' ***********                          [332]
1310 ' * MONSTRE *                          [331]
1320 ' ***********                          [332]
1330 DI                                     [84]
1340 IF X(X1,Y1)<>0 THEN X1=XP:Y1=Y [8230]
P:SC=SC+500:X(X1,Y1)=0:CALL &A000,F
NPO(X1,Y1),SP(10):CALL &A000,FNPO(X
1,Y1),SP(8):MON=MSTR:GOSUB 1490:GOT
O 1400
1350 IF X1<X THEN IF X(X1+1,Y1)=0 T [5679]
HEN CALL &9400,FNPO(X1,Y1),SP(8):X1
=X1+1:GOTO 1400
1360 IF X1>X THEN IF X(X1-1,Y1)=0 T [4506]
HEN CALL &9500,FNPO(X1,Y1),SP(8):X1
=X1-1:GOTO 1400
1370 IF Y1<Y THEN IF X(X1,Y1+1)=0 T [3300]
HEN CALL &9700,FNPO(X1,Y1),SP(8):Y1
=Y1+1:GOTO 1400
1380 IF Y1>Y THEN IF X(X1,Y1-1)=0 T [3666]
HEN CALL &9600,FNPO(X1,Y1),SP(8):Y1
=Y1-1
1390 IF X=X1 AND Y=Y1 THEN FIN=1        [1266]
1400 EI:RETURN                              [416]
1410 ' **********                           [315]
1420 ' * VIE -1 +                           [387]
1430 ' **********                           [315]
1440 MU=REMAIN(0)                           [333]
1450 LOCATE 1,1:PEN 6:PRINT"VIE:";      [1197]
1460 PEN 9:VIE=VIE-1:PRINT VIE          [1666]
1470 IF VIE=0 THEN GOTO 2400            [791]
1480 GOTO 1770                              [361]
1490 ' *********                            [418]
1500 ' * SCORE *                            [83]
1510 ' *********                            [418]
1520 DI:PEN 7:LOCATE 13,1:PRINT USI [2347]
NG"######";SC:
1530 EI:RETURN                              [416]
1540 ' **************************           [894]
1550 ' * TABLEAU INTERMEDIAIRE *           [1908]
1560 ' **************************           [894]
1570 PAC=-1                                 [392]
1580 CLS#1                                  [373]
1590 FOR I=1 TO 10                          [315]
1600 READ A                                 [428]
1610 B$=BIN$(A,16)                          [314]
1620 FOR O=1 TO 16                          [956]
1630 B1=VAL(MID$(B$,O,1))                   [917]
1640 IF B1=1 THEN CALL &A000,FNPO(O [2460]
,I),SP(1) ELSE CALL &A000,FNPO(O,I)
,SP(9):PAC=PAC+1
1650 IF B1=0 THEN B1=2                      [1139]
1660 X(O,I)=B1                              [684]
1670 NEXT:NEXT                              [1022]
1680 X(1,10)=0                              [899]
1690 X=1:Y=10                               [1012]
1700 CALL &A000,FNPO(1,10),SP(4)        [1081]
1710 IF INKEY(GA)=0 AND X>1 AND X(X [11672]
-1,Y)<>1 THEN CALL &9500,FNPO(X,Y),
SP(5):X=X-1:IF X(X,Y)=2 THEN PAC=PA
C-1:X(X,Y)=0:SC=SC+25:SOUND 1,35,15
,15,2:GOSUB 1490:GOTO 1750 ELSE GOT
O 1750
1720 IF INKEY(DA)=0 AND X<16 AND X( [8026]
X+1,Y)<>1 THEN CALL &9400,FNPO(X,Y)
,SP(4):X=X+1:IF X(X,Y)=2 THEN PAC=P
AC-1:X(X,Y)=0:SC=SC+25:SOUND 1,35,1
5,15,2:GOSUB 1490:GOTO 1750 ELSE GO
TO 1750
1730 IF INKEY(HT)=0 AND Y>1 AND X(X [10002]
,Y-1)<>1 THEN CALL &9600,FNPO(X,Y),
SP(7):Y=Y-1:IF X(X,Y)=2 THEN PAC=PA
C-1:X(X,Y)=0:SC=SC+25:SOUND 1,35,15
,15,2:GOSUB 1490:GOTO 1750 ELSE GOT
O 1750
1740 IF INKEY(BA)=0 AND Y<10 AND X( [8510]
X,Y+1)<>1 THEN CALL &9700,FNPO(X,Y)
,SP(6):Y=Y+1:IF X(X,Y)=2 THEN PAC=P
AC-1:X(X,Y)=0:SC=SC+25:SOUND 1,35,1
5,15,2:GOSUB 1490:GOTO 1750 ELSE GO
TO 1750
1750 IF PAC=0 THEN TA=TA+1:FOR I=20 [4716]
0 TO 0 STEP -20:SOUND 1,I,4,15:NEXT
:GOTO 1770
```

```
1760 GOTO 1710                          [357]
1770 ' **************                   [445]
1780 ' * TABLEAU +1 *                   [497]
1790 ' **************                   [445]
1800 MU=REMAIN(0)                       [333]
1810 IF TA=26 THEN TA=1                 [1537]
1820 ON TA GOTO 1830,1840,1850,1860     [7765]
,2030,1870,1880,1890,1900,2030,1910
,1920,1930,1940,2030,1950,1960,1970
,1980,2030,1990,2000,2010,2020,2030
1830 RESTORE 3000:GOTO 2040 ' TABLE     [2037]
AU 1
1840 RESTORE 3030:GOTO 2040 ' TABLE     [1867]
AU 2
1850 RESTORE 3060:GOTO 2040 ' TABLE     [1290]
AU 3
1860 RESTORE 3090:GOTO 2040 ' TABLE     [2353]
AU 4
1870 RESTORE 3120:GOTO 2040 ' TABLE     [1219]
AU 6
1880 RESTORE 3150:GOTO 2040 ' TABLE     [1866]
AU 7
1890 RESTORE 3180:GOTO 2040 ' TABLE     [2260]
AU 8
1900 RESTORE 3210:GOTO 2040 ' TABLE     [2168]
AU 9
1910 RESTORE 3240:GOTO 2040 ' TABLE     [1592]
AU 11
1920 RESTORE 3270:GOTO 2040 ' TABLE     [2038]
AU 12
1930 RESTORE 3300:GOTO 2040 ' TABLE     [1441]
AU 13
1940 RESTORE 3330:GOTO 2040 ' TABLE     [1620]
AU 14
1950 RESTORE 3360:GOTO 2040 ' TABLE     [1083]
AU 16
1960 RESTORE 3390:GOTO 2040 ' TABLE     [1644]
AU 17
1970 RESTORE 3420:GOTO 2040 ' TABLE     [2119]
AU 18
1980 RESTORE 3450:GOTO 2040 ' TABLE     [1651]
AU 19
1990 RESTORE 3480:GOTO 2040 ' TABLE     [2404]
AU 21
2000 RESTORE 3510:GOTO 2040 ' TABLE     [1351]
AU 22
2010 RESTORE 3540:GOTO 2040 ' TABLE     [1674]
AU 23
2020 RESTORE 3570:GOTO 2040 ' TABLE     [1626]
AU 24
2030 RESTORE 3600:GOTO 1540 ' TABLE     [3380]
AUX INTERMEDIAIRES 5,10,15,20 ET 25
2040 CLS#1:GOSUB 2050:GOTO 1110         [1856]
2050 ' ***********************          [577]
2060 ' * DESSIN D'UN TABLEAU *          [753]
2070 ' ***********************          [577]
2080 MU=REMAIN(0)                       [333]
2090 FOR I=1 TO 10:READ A:A$(I)=BIN     [3048]
$(A,16):NEXT
2100 FOR I=1 TO 10                      [315]
2110 FOR O=1 TO 16                      [956]
2120 B1=VAL(MID$(A$(I),O,1))            [783]
2130 X(O,I)=0                           [734]
2140 IF B1=1 THEN CALL &A000,FNPO(O     [1248]
,I),SP(2)
2150 IF B1=1 THEN X(O,I)=2              [940]
2160 NEXT:NEXT                          [1022]
2170 FOR I=1 TO 4                       [450]
2180 READ X,Y                           [469]
2190 CALL &A000,FNPO(X,Y),SP(3)         [1449]
2200 X(X,Y)=3                           [669]
2210 NEXT                               [350]
2220 READ XB,YB                         [1068]
2230 CALL &A000,FNPO(XB,YB),SP(1)       [975]
2240 X(XB,YB)=1                         [890]
2250 READ X1,Y1:XP=X1:YP=Y1             [2205]
2260 CALL &A000,FNPO(X1,Y1),SP(8)       [638]
2270 READ X,Y                           [469]
2280 CALL &A000,FNPO(X,Y),SP(4)         [1459]
2290 READ MSTR                          [535]
2300 RETURN                             [555]
2310 ' *************************        [768]
2320 ' * UNE BRIQUE DISPARAIT *         [1424]
2330 ' *************************        [768]
2340 DI                                 [84]
2350 XD=INT(RND*16)+1                   [1637]
2360 YD=INT(RND*10)+1                   [1317]
2370 IF X(XD,YD)=2 THEN CALL &A000,     [5817]
FNPO(XD,YD),SP(10):SOUND 1,35,45,15
,1,1:X(XD,YD)=0 ELSE GOTO 2350
2380 EI                                 [86]
2390 RETURN                             [555]
2400 ' ****************                 [528]
2410 ' * PRESENTATION *                 [1606]
2420 ' ****************                 [528]
2430 MODE 0:PEN 7                       [1133]
2440 PRINT"         JEASON"             [1007]
2450 PEN 2                              [548]
2460 PRINT"         ------"             [1134]
2470 LOCATE 1,5                         [623]
2480 PRINT" ⊕11 04-06 REGLES"           [1389]
2490 PRINT:PRINT                        [743]
2500 PRINT" ⊕12 04-06 JOYSTICK"         [1181]
2510 PRINT:PRINT                        [743]
2520 PRINT" ⊕13 04-06 CLAVIER"          [1823]
2530 F$=".   .   .   .   .   JEASON     [5203]
- (C)copyright R.O.S. by SEBASTIEN
 ROYER en Novembre 1988"
2540 LO=1:PEN 8                         [856]
2550 LOCATE 1,25:PRINT LEFT$(F$,20)     [1857]
;
2560 F$=RIGHT$(F$,LEN(F$)-1)+LEFT$(     [2237]
F$,1)
2570 A$=INKEY$                          [278]
2580 IF A$="1" THEN GOTO 2630           [1326]
2590 IF A$="2" THEN AB=18:GA=74:DA=     [3868]
75:HT=72:BA=73:GOTO 920
2600 IF A$="3" THEN AB=18:GA=8:DA=1     [3138]
:HT=0:BA=2:GOTO 920
2610 CALL &BD19                         [352]
2620 GOTO 2550                          [365]
2630 ' ****************                 [528]
2640 ' **** REGLES ****                 [367]
2650 ' ****************                 [528]
2660 MODE 1:INK 0,0:INK 1,2:INK 2,6     [2759]
:INK 3,24:BORDER 0:PEN 3
2670 PRINT"                 REGLES"     [1292]
2680 PEN 2                              [548]
2690 PRINT"                 ======"     [1302]
2700 PEN 1                              [549]
2710 PRINT                              [361]
2720 PRINT"  Vous etes JEASON (le b     [3940]
onhomme rouge)"
2730 PRINT"et vous devez reunir les     [4486]
 4 briques de"
2740 PRINT"feu autour de la boule n     [2274]
oire."
2750 PRINT                              [361]
2760 PRINT"  Pour cela, vous pouvez     [4103]
 deplacer les"
```

```
2770 PRINT"briques ou la boule en l [5512]
es poussant"
2780 PRINT"(une par une)."           [397]
2790 PRINT                           [361]
2800 PRINT"  Mais les choses se gat [3241]
tent, car un"
2810 PRINT"monstre redoutable prote [5946]
ge les lieux."
2820 PRINT"Neanmoins, rassurez-vous [4912]
; vous pourrez"
2830 PRINT"le tuer en l'ecrasant av [4327]
ec quelques"
2840 PRINT"briques ou bien la boule [2609]
."
2850 PRINT                           [361]
2860 PRINT"  Tout les 5 tableaux (i [4576]
l y en a 25),"
2870 PRINT"vous pourrez gagner enor [4051]
mement de"
2880 PRINT"points en mangeant les r [3392]
ubis qui rem-"
2890 PRINT"-plissent l'ecran."      [1297]
2900 PRINT                           [361]
2910 PRINT"  Voila, vous savez tout [3370]
."
2920 PRINT"   Et je n'est qu'une ch [3263]
ose a vous"
2930 PRINT"    dire : ";:PEN 2:PRIN [3212]
T"BONNE CHANCE"
2940 PEN 3:LOCATE 32,25:PRINT"[TOUC [5574]
HE]":CALL &BB06:INK 0,2:INK 1,1:INK
 2,0:INK 3,3
2950 GOTO 2400                       [347]
2960 ' ************                  [502]
2970 ' * TABLEAUX *                  [630]
2980 ' ************                  [502]
2990 ' TAB 1                         [94]
3000 DATA 0,8192,10210,8210,2056,20 [2023]
50,2160,0,-1568,0
3010 DATA 4,3,8,6,13,4,14,9,9,4,1,1 [1447]
0,16,1,10
3020 ' TAB 2                         [93]
3030 DATA 0,16352,4,20996,17140,204 [2392]
5,16900,516,14368,32
3040 DATA 2,2,4,10,13,2,13,9,9,7,16 [1523]
,4,4,3,10
3050 ' TAB 3                         [92]
3060 DATA 4080,4104,10212,16386,327 [2464]
66,16386,16386,16386,32766,0
3070 DATA 4,7,7,7,10,7,13,7,9,2,3,4 [1272]
,1,10,9
3080 ' TAB 4                         [91]
3090 DATA 0,896,1088,-31672,-15736, [2468]
-7928,8,15375,0,0
3100 DATA 2,9,4,2,8,4,15,3,9,9,16,7 [1373]
,1,3,9
3110 ' TAB 6                         [89]
3120 DATA 4,392,584,13932,8452,8580 [1714]
,12684,961,0,4
3130 DATA 3,9,8,4,12,9,15,5,3,2,14, [1885]
2,6,10,8
3140 ' TAB 7                         [88]
3150 DATA 0,-2721,0,0,32766,0,0,-1, [1480]
-15292,4368
3160 DATA 3,3,4,6,10,1,14,6,8,9,1,1 [1659]
,1,10,8
3170 ' TAB 8                         [119]
3180 DATA -4370,10408,11502,-22366, [2492]
20142,0,16654,16640,32246,0
3190 DATA 2,4,5,7,12,4,15,8,10,7,1, [1897]
10,16,10,7
3200 ' TAB 9                         [118]
3210 DATA 8576,962,966,-32320,24720 [1829]
,24768,192,448,29070,768
3220 DATA 3,2,7,6,13,6,15,8,14,7,2, [2331]
4,14,2,7
3230 ' TAB 11                        [271]
3240 DATA 0,31902,4434,4434,-28334, [2832]
-28526,0,6168,960,0
3250 DATA 4,1,6,9,9,4,15,9,2,9,1,3, [1542]
15,1,6
3260 ' TAB 12                        [266]
3270 DATA 0,0,15196,11218,15058,108 [1951]
44,0,2064,2096,0
3280 DATA 3,9,6,2,11,2,14,9,8,9,11, [1557]
7,1,1,6
3290 ' TAB 13                        [269]
3300 DATA 18400,8194,4356,2056,1371 [2237]
2,2476,4112,8328,16388,2018
3310 DATA 5,5,7,8,10,3,12,6,6,8,1,1 [1547]
0,13,1,5
3320 ' TAB 14                        [264]
3330 DATA 24,16,10772,10772,10772,1 [2428]
0773,10868,8822,15348,0
3340 DATA 4,7,9,5,11,5,13,5,10,5,8, [1348]
5,16,1,2,1,5
3350 ' TAB 16                        [262]
3360 DATA 4624,528,4688,4160,4160,1 [2939]
6382,-16391,-32767,-32456,-16387
3370 DATA 4,8,6,8,8,8,15,8,14,4,2,1 [1253]
0,1,3,4
3380 ' TAB 17                        [265]
3390 DATA 0,0,-8739,0,30070,0,32511 [1845]
,3072,-3,0
3400 DATA 5,2,9,2,7,4,7,1,7,3,16,1, [1782]
1,10,4
3410 ' TAB 18                        [260]
3420 DATA 0,-8976,0,644,584,560,584 [2260]
,644,16896,512
3430 DATA 3,2,3,5,14,6,13,9,10,8,12 [2311]
,4,8,5,11,4,3
3440 ' TAB 19                        [263]
3450 DATA -31743,-31737,-29187,-284 [4003]
16,-2714,17768,16651,14713,-32507,1
7941
3460 DATA 3,3,3,7,7,4,11,9,6,8,10,4 [1291]
,1,10,11,4,3
3470 ' TAB 21                        [265]
3480 DATA 768,28674,21474,20482,206 [2721]
10,22174,29284,1153,0,-3746
3490 DATA 12,8,12,2,5,2,3,5,14,8,7, [1951]
5,1,1,6,5,2
3500 ' TAB 22                        [272]
3510 DATA 1056,31710,19024,19024,19 [2867]
032,29254,21058,19010,19422,1056
3520 DATA 3,4,9,4,9,7,6,6,12,8,1,1, [2116]
3,5,2,1,2
3530 ' TAB 23                        [271]
3540 DATA -32768,-24608,-32744,-250 [2854]
62,17050,16794,16410,4058,24,4088
3550 DATA 3,3,8,5,14,8,15,2,3,8,8,4 [1477]
,1,10,2
3560 DATA ' TAB 24                   [308]
3570 DATA 0,8192,8204,8204,8204,218 [2634]
45,-21845,21845,-21845,21845
3580 DATA 2,3,7,6,11,6,15,3,8,3,1,5 [2178]
,1,10,2
3590 ' TAB INTERMEDIAIRE             [1163]
3600 DATA 0,0,-196,0,-2068,0,32767, [1462]
0,-2,0
```

JEU

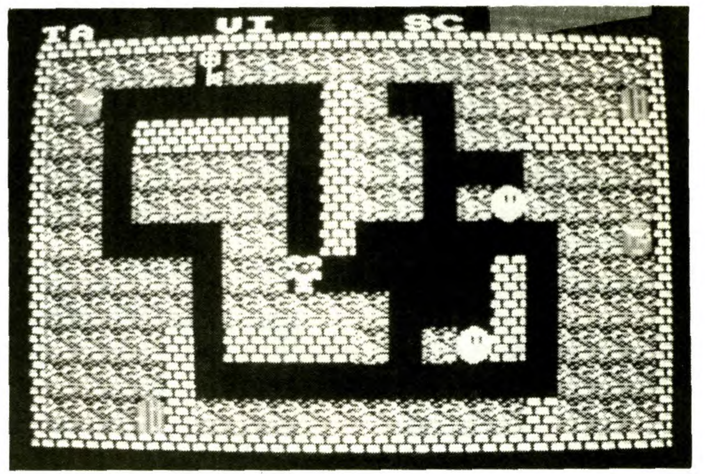
Ce programme figure sur la disquette : HORS-SERIE N° 10

PING & PONG CONSTRUCTION KIT

Egarées dans un univers étrange qui ne leur est pas coutumier, deux balles de ping-pong devront rivaliser d'astuce pour rejoindre à temps le lieu du championnat.
L'astuce de ce jeu somme toute classique (la quête de clefs ouvrant des passages) réside dans le couplage de deux personnages. Le passage sur certains objets présents dans le décor, facilite ou complique la tâche de nos héros. Un éditeur de tableaux (cinq tableaux d'origine) présent dans le programme, permet d'être l'artisan de son propre calvaire. Les règles du jeu et le mode d'emploi de l'éditeur sont inclus.

Sauvegarde

Sauvez tout d'abord sous un nom de votre choix (PINGPONG), le programme Basic principal. Entrez ensuite par *Amsaisie* (reportez-vous à son mode d'emploi) les codes hexadécimaux du second listing. Spécifiez 96EC comme adresse de début et sauvez le langage machine par l'option « S » sous le nom « PING ». Si vous ne souhaitez pas saisir en une seule fois la totalité des codes, morcelez votre travail en créant plusieurs fichiers (PING1, PING2, etc.). Ces derniers devront ultérieurement être chargés à la suite (LOAD"PING1.BIN" : LOAD"PING2.BIN", etc.) après un MEMORY &96EB et sauvegardés ainsi dans un fichier unique :
SAVE"PING",b,&96EC,&AA0



Claude Le Moullec

```
10 REM  ::::::::::::::::::::::::         [1736]
20 REM  :                      :         [419]
30 REM  : P I N G  &  P O N G :          [1460]
40 REM  :                      :         [419]
50 REM  :     CONSTRUCTION     :         [1196]
60 REM  :                      :         [419]
70 REM  :         KIT          :         [680]
80 REM  :                      :         [419]
90 REM  :  Claude Le Moullec   :         [2031]
100 REM :                      :         [419]
110 REM ::::::::::::::::::::::::         [1736]
120 SYMBOL AFTER 240                     [1392]
130 SYMBOL 241,0,138,142,138,138,13 [2116]
8,234,0
140 SYMBOL 242,0,224,128,128,128,12 [2607]
8,224,0
150 SYMBOL 243,0,238,138,138,234,42 [2714]
,238,0
160 SYMBOL 244,0,238,132,132,196,13 [1551]
2,132,0
170 SYMBOL 245,0,174,170,170,174,23 [2677]
4,170,0
180 SYMBOL 246,0,238,168,232,204,16 [2418]
8,174,0
190 MEMORY &96EB                         [724]
200 LOAD "!ping",&96EC                   [647]
210 REM ::::::::::::::::::::::::         [1736]
220 REM :                      :         [419]
230 REM :  VARIABLES DE BASE   :         [2081]
240 REM :                      :         [419]
250 REM ::::::::::::::::::::::::         [1736]
260 CALL &BBFF:MODE 0:BORDER 0           [1633]
270 DEFINT a-y:POKE &A024,0              [1043]
280 INK 0,0:INK 1,24:INK 2,13:INK 3 [6927]
,16:INK 4,26:INK 5,18:INK 6,6:INK 7
,15:INK 8,25:INK 9,7:INK 10,3:INK 1
1,11:INK 12,2:INK 13,9
290 DIM sp(16):FOR h=0 TO 15:sp(h+1 [3146]
)=&96EC+(h*64):NEXT
300 FOR h=0 TO 6:tabl(h+1)=&9AEC+(h [2237]
*209):NEXT
310 DIM jeu(21,13):pg=3:ta=1             [1015]
320 DEF FN po(a,b)=&C002+(b-1)*80+( [1226]
a-1)*4
330 DEF FN je(c,d)=jeu(c+1,(d+1)/2) [1056]
340 TR$=CHR$(22)+CHR$(1)                 [806]
350 NR$=CHR$(22)+CHR$(0)                 [700]
360 LMC$=CHR$(241)+CHR$(242)+CHR$(2 [3997]
43)+CHR$(244)+CHR$(245)+CHR$(246)
370 GOSUB 2470:GOTO 2810                 [1417]
380 jx=1:kx=1:jy=2:ky=2                  [1192]
390 VIE=5:SC=0                           [896]
400 ENT 1,100,2,2:ENV 1,100,3,1          [1188]
410 REM ::::::::::::::::::::::::         [1736]
420 REM :                      :         [419]
430 REM :      TABLEAU +1      :         [574]
440 REM :                      :         [419]
450 REM ::::::::::::::::::::::::         [1736]
460 CALL &A039,&C050,sp(2)               [1433]
470 FOR h=1 TO 13:FOR g=1 TO 21:jeu [1939]
(g,h)=2:NEXT g,h
480 LOCATE 1,1:PRINT SPACE$(20)          [1907]
490 LOCATE 1,1:PEN 4:PRINT "TA    V [4369]
I    SC";:PEN 6:LOCATE 16,1:PRINT"0
0000"
500 LOCATE 3,1:PRINT TA:LOCATE 9,1: [3708]
PRINT VIE:GOSUB 2380
510 ON ta GOTO 520,540,560,590,620, [1056]
```

```
650
520 CALL &A0A8,tabl(ta),sp(1)                [1646]
530 x=7:y=9:x1=11:y1=21:px=1:py=23:          [3634]
px1=19:py1=23:GOTO 750
540 CALL &A0A8,tabl(ta),sp(1)                [1646]
550 x=7:y=23:x1=15:y1=23:px=1:py=3:          [3087]
px1=19:py1=23:GOTO 750
560 CALL &A0A8,tabl(ta),sp(1)                [1646]
570 x=12:y=5:x1=11:y1=13:px=19:py=5          [4659]
:px1=4:py1=23:mx=3:my=5:m1=1:sens=1
580 GOTO 750                                 [423]
590 CALL &A0A8,tabl(ta),sp(1)                [1646]
600 x=17:y=13:x1=12:y1=19:px=19:py=          [4641]
3:px1=5:py1=5:mx=1:my=3:m1=1:sens=1
610 GOTO 750                                 [423]
620 CALL &A0A8,tabl(ta),sp(1)                [1646]
630 x=6:y=9:x1=11:y1=17:px=19:py=5:          [3229]
px1=2:py1=19:mx=3:my=5:m1=1:sens=1
640 GOTO 750                                 [423]
650 IF PEEK(tabl(7)+31)=1 THEN 670           [1707]
ELSE ta=1:GOTO 380
660 PEN 6:LOCATE 4,11:PRINT"PAS DE           [5077]
TABLEAU":WHILE INKEY$="":WEND:TA=1:
CLS:GOTO 380
670 CALL &A0A8,tabl(ta),sp(1)                [1646]
680 x=PEEK(tabl(7)+20):y=PEEK(tabl(          [3579]
7)+21)
690 x1=PEEK(tabl(7)+22):y1=PEEK(tab          [2419]
l(7)+23)
700 px=PEEK(tabl(7)+24):py=PEEK(tab          [1874]
l(7)+25)
710 px1=PEEK(tabl(7)+26):py1=PEEK(t          [1784]
abl(7)+27)
720 IF PEEK(tabl(7)+28)=0 THEN 740           [2177]
730 m1=PEEK(tabl(7)+28):mx=PEEK(tab          [3420]
l(7)+29):my=PEEK(tabl(7)+30):sens=1
740 GOTO 750                                 [423]
750 d=0:FOR g=1 TO 11:FOR h=1 TO 19          [2481]
760 a=PEEK(tabl(ta)+d):jeu(h+1,g+1)          [2320]
=a
770 d=d+1:NEXT h,g:SOUND 1,250,40,1          [2537]
5,,,12
780 CALL &A00D,FN po(x1,y1),sp(5)            [1606]
790 REM :::::::::::::::::::::::              [1736]
800 REM :                     :              [419]
810 REM :  ROUTINE PRINCIPALE :              [2225]
820 REM :                     :              [419]
830 REM :::::::::::::::::::::::              [1736]
840 CALL &A00D,FN po(x,y),sp(5)              [1641]
850 CALL &A00D,FN po(x1,y1),sp(5)            [1606]
860 IF m1=1 THEN 1900                        [289]
870 IF INKEY(ha)=0 THEN 940                  [968]
880 IF INKEY(ba)=0 THEN 980                  [1742]
890 IF INKEY(ga)=0 THEN 1020                 [1017]
900 IF INKEY(da)=0 THEN 1060                 [1175]
910 IF INKEY(fe)=0 THEN 2260                 [1100]
920 GOTO 860                                 [536]
930 REM ::::: HAUT :::::                     [1010]
940 IF pa1=0 THEN dx=0:dy=-jy ELSE           [1993]
dx=0:dy=0
950 IF pa2=0 THEN fx=0:fy=-ky ELSE           [3500]
fx=0:fy=0
960 GOTO 1140                                [311]
970 REM ::::: BAS  :::::                     [1020]
980 IF pa1=0 THEN dx=0:dy=jy ELSE d          [2387]
x=0:dy=0
990 IF pa2=0 THEN fx=0:fy=ky ELSE f          [2939]
x=0:fy=0
1000 GOTO 1140                               [311]
1010 REM ::::: GAUC :::::                    [1106]
1020 IF pa1=0 THEN dx=-jx:dy=0:pg=4          [2031]
 ELSE dx=0:dy=0
1030 IF pa2=0 THEN fx=-kx:fy=0 ELSE          [3154]
 fx=0:fy=0
1040 GOTO 1140                               [311]
1050 REM ::::: DROI :::::                    [1191]
1060 IF pa1=0 THEN dx=jx:dy=0:pg=3           [1300]
ELSE dx=0:dy=0
1070 IF pa2=0 THEN fx=kx:fy=0 ELSE           [2860]
fx=0:fy=0
1080 GOTO 1140                               [311]
1090 REM :::::::::::::::::::::::             [1736]
1100 REM :                     :             [419]
1110 REM : PING & PONG BOUGENT :             [1906]
1120 REM :                     :             [419]
1130 REM :::::::::::::::::::::::             [1736]
1140 IF pa1=1 THEN 1320 ELSE IF pa2          [2571]
=1 THEN 1250
1150 x=x+dx:y=y+dy:x1=x1+fx:y1=y1+f          [1911]
y
1160 IF FN je(x,y)<2 AND FN je(x1,y          [2988]
1)<2 THEN 1170 ELSE 1430
1170 SOUND 1,300,5,1,1,1,15                  [1467]
1180 CALL &A00D,FN po(x-dx,y-dy),sp          [4294]
(15):jeu(x-dx+1,(y+1-dy)/2)=0
1190 CALL &A00D,FN po(x1-fx,y1-fy),          [4783]
sp(15):jeu(x1-fx+1,(y1+1-fy)/2)=0
1200 CALL &A00D,FN po(x,y),sp(pg+2)          [2354]
:jeu(x+1,(y+1)/2)=3
1210 CALL &A00D,FN po(x1,y1),sp(pg+          [3823]
2):jeu(x1+1,(y1+1)/2)=4
1220 CALL &A00D,FN po(x1,y1),sp(pg)          [1986]
1230 CALL &A00D,FN po(x,y),sp(pg):G          [1719]
OTO 860
1240 REM :::: PING BOUGE SEUL ::::           [1679]
1250 x=x+dx:y=y+dy                           [1457]
1260 IF FN je(x,y)<2 THEN 1270 ELSE          [1231]
 1450
1270 SOUND 1,300,5,1,1,1,15                  [1467]
1280 CALL &A00D,FN po(x-dx,y-dy),sp          [4294]
(15):jeu(x-dx+1,(y+1-dy)/2)=0
1290 CALL &A00D,FN po(x,y),sp(pg+2)          [2354]
:jeu(x+1,(y+1)/2)=3
1300 CALL &A00D,FN po(x,y),sp(pg):G          [1719]
OTO 860
1310 REM :::: PONG BOUGE SEUL ::::           [1113]
1320 x1=x1+fx:y1=y1+fy                       [971]
1330 IF FN je(x1,y1)<2 THEN 1340 EL          [1087]
SE 1460
1340 SOUND 1,300,5,1,1,1,15                  [1467]
1350 CALL &A00D,FN po(x1-fx,y1-fy),          [4783]
sp(15):jeu(x1-fx+1,(y1+1-fy)/2)=0
1360 CALL &A00D,FN po(x1,y1),sp(pg+          [3823]
2):jeu(x1+1,(y1+1)/2)=4
1370 CALL &A00D,FN po(x1,y1),sp(pg)          [2551]
:GOTO 860
1380 REM :::::::::::::::::::::::             [1736]
1390 REM :                     :             [419]
1400 REM : LES DIVERS CONTACTS :             [671]
1410 REM :                     :             [419]
1420 REM :::::::::::::::::::::::             [1736]
1430 IF FN je(x,y)=2 OR FN je(x1,y1          [1993]
)=2 THEN 1570
1440 IF FN je(x,y)<2 THEN 1460 ELSE          [1181]
 1450
1450 ON FN je(x,y) GOTO 860,1570,15          [4334]
70,1570,2260,1570,1670,1550,1580,16
00,1490,1720
```

```
1460 ON FN je(x1,y1) GOTO 860,1570,  [3840]
1570,1570,2260,1570,1670,1560,1510,
1530,1490,1790
1470 jx=-1:GOTO 1170                  [552]
1480 jy=-2:GOTO 1170                  [745]
1490 jx=1:kx=1:jy=2:ky=2:IF pa1=0 A  [2481]
ND pa2=0 THEN 1170
1500 IF pa1=1 THEN 1340 ELSE 1270    [1720]
1510 kx=-1:IF pa1=0 AND pa2=0 THEN   [3526]
1170.
1520 IF pa1=1 THEN 1340 ELSE 1270    [1720]
1530 ky=-2:IF pa1=0 AND pa2=0 THEN   [2694]
1170
1540 IF pa1=1 THEN 1340 ELSE 1270    [1720]
1550 IF pp2=1 THEN 1270 ELSE pa1=1:  [1956]
INK 1,22:GOTO 3230
1560 IF pp1=1 THEN 1340 ELSE pa2=1:  [2432]
INK 1,22:GOTO 3230
1570 x=x-dx:y=y-dy:x1=x1-fx:y1=y1-f  [2994]
y:GOTO 860
1580 jx=-1:IF pa1=0 AND pa2=0 THEN   [1146]
1170
1590 IF pa1=1 THEN 1340 ELSE 1270    [1720]
1600 jy=-2:IF pa1=0 AND pa2=0 THEN   [987]
1170
1610 IF pa1=1 THEN 1340 ELSE 1270    [1720]
1620 REM ::::::::::::::::::::::      [1736]
1630 REM :                    :      [419]
1640 REM :     LES PORTES     :      [689]
1650 REM :                    :      [419]
1660 REM ::::::::::::::::::::::      [1736]
1670 CALL &A00D,FN po(px,py),sp(8)   [1685]
1680 CALL &A00D,FN po(px1,py1),sp(8  [1493]
)
1690 jeu(px+1,(py+1)/2)=12:jeu(px1+  [2552]
1,(py1+1)/2)=12
1700 port=1:SOUND 1,2056,60,15:SC=S  [3045]
C+10:GOSUB 2380:GOTO 1170
1710 REM :::: PING PASSE ::::        [957]
1720 CALL &A00D,FN po(x-dx,y-dy),sp  [4294]
(15):jeu(x-dx+1,(y+1-dy)/2)=0
1730 CALL &A00D,FN po(x,y),sp(9):je  [2838]
u(x+1,(y+1)/2)=6
1740 SOUND 1,2056,60,15:pp1=1        [2257]
1750 DI:mu=REMAIN(3):pa1=1:pa2=0:IN  [2398]
K 1,24:EI
1760 pass=pass+1:IF pass=2 THEN 217  [1305]
0
1770 SC=SC+20:GOSUB 2380:GOTO 1330   [965]
1780 REM :::: PONG PASSE ::::        [642]
1790 CALL &A00D,FN po(x1-fx,y1-fy),  [4783]
sp(15):jeu(x1-fx+1,(y1+1-fy)/2)=0
1800 CALL &A00D,FN po(x1,y1),sp(9):  [3325]
jeu(x1+1,(y1+1)/2)=6
1810 SOUND 1,2056,60,15:pp2=1        [2277]
1820 DI:mu=REMAIN(3):pa1=0:pa2=1:IN  [3132]
K 1,24:EI
1830 pass=pass+1:IF pass=2 THEN 217  [1305]
0
1840 SC=SC+20:GOSUB 2380:GOTO 1260   [1178]
1850 REM ::::::::::::::::::::::      [1736]
1860 REM :                    :      [419]
1870 REM :   LE PETIT DIABLE  :      [397]
1880 REM :                    :      [419]
1890 REM ::::::::::::::::::::::      [1736]
1900 mx1=mx:my1=my                   [592]
1910 ON sm GOTO 1920,2000,2040,2080  [2468]
1920 mx=mx+1:IF FN je(mx,my)=3 OR F  [3477]
N je(mx,my)=4 THEN 2260
1930 IF FN je(mx,my)=0 THEN 1960     [1212]
1940 mx=mx-1                         [764]
1950 sm=INT(RND*4)+1:IF sm=1 OR sm=  [3128]
3 THEN 1950 ELSE 870
1960 CALL &A00D,FN po(mx1,my1),sp(1  [1809]
5)
1970 jeu(mx1+1,(my1+1)/2)=0          [1057]
1980 CALL &A00D,FN po(mx,my),sp(7)   [1971]
1990 jeu(mx+1,(my+1)/2)=5:GOTO 870   [966]
2000 my=my+2:IF FN je(mx,my)=3 OR F  [4269]
N je(mx,my)=4 THEN 2260
2010 IF FN je(mx,my)=0 THEN 1960     [1212]
2020 my=my-2                         [542]
2030 sm=INT(RND*4)+1:IF sm=2 OR sm=  [1609]
4 THEN 2030 ELSE 870
```




```
2040 mx=mx-1:IF FN je(mx,my)=3 OR F  [2655]
N je(mx,my)=4 THEN 2260
2050 IF FN je(mx,my)=0 THEN 1960     [1212]
2060 mx=mx+1                         [768]
2070 sm=INT(RND*4)+1:IF sm=3 OR sm=  [3150]
1 THEN 2070 ELSE 870
2080 my=my-2:IF FN je(mx,my)=3 OR F  [1475]
N je(mx,my)=4 THEN 2260
2090 IF FN je(mx,my)=0 THEN 1960     [1212]
2100 my=my+2                         [538]
2110 sm=INT(RND*4)+1:IF sm=4 OR sm=  [3211]
2 THEN 2110 ELSE 870
2120 REM ::::::::::::::::::::::      [1736]
2130 REM :                    :      [419]
2140 REM :    GAGNE/PERDU     :      [1324]
2150 REM :                    :      [419]
2160 REM ::::::::::::::::::::::      [1736]
2170 FOR h=0 TO 3:mu=REMAIN(h):NEXT  [2096]
2180 FOR H=1 TO 16:INK 0,H:SC=SC+5:  [2504]
GOSUB 2380
2190 SOUND 1,H*15,10,14:FOR T=1 TO   [2846]
200:NEXT T,H:INK 0,0
2200 ta=ta+1:IF ta=7 THEN ta=1       [1921]
2210 ERASE jeu:DIM jeu(21,13)        [695]
2220 pass=0:pa1=0:pa2=0:INK 1,24     [2028]
2230 jx=1:kx=1:jy=2:ky=2             [1192]
2240 m1=0:pp1=0:pp2=0:GOTO 460       [2466]
2250 REM ::::      PERDU     ::::    [754]
2260 FOR h=0 TO 3:mu=REMAIN(h):NEXT  [2096]
```

```
2270 vie=vie-1:BORDER 26:INK 0,26:I [1933]
NK 1,26:INK 2,26
2280 OUT &BC00,2:OUT &BC49,49:SOUND [4228]
 4,1500,50,7,0,0,10:OUT &BC00,2:OUT
 &BC49,46
2290 FOR t=1 TO 4000:NEXT           [1133]
2300 BORDER 0:INK 0,0:INK 1,24:INK  [2769]
2,13
2310 IF vie=0 THEN RUN 260          [868]
2320 GOTO 2210                      [361]
2330 REM ::::::::::::::::::::::     [1736]
2340 REM :                    :     [419]
2350 REM :   GESTION DU SCORE  :    [1698]
2360 REM :                    :     [419]
2370 REM ::::::::::::::::::::::     [1736]
2380 PEN 0:IF SC>999 THEN LOCATE 16 [6340]
,1:PRINT CHR$(143)+(143)+CHR$(143)+
CHR$(143):PEN 6:LOCATE 17,1:PRINT T
R$;SC;NR$:RETURN
2390 IF SC>99 THEN LOCATE 17,1:PRIN [7812]
T CHR$(143)+CHR$(143)+CHR$(143):PEN
 6:LOCATE 16,1:PRINT TR$;SC;NR$:RET
URN
2400 IF SC>9 THEN LOCATE 18,1:PRINT [5558]
 CHR$(143)+CHR$(143):PEN 6:LOCATE 1
7,1:PRINT TR$;SC;NR$:RETURN
2410 LOCATE 19,1:PRINT CHR$(143):PE [3695]
N 6:LOCATE 18,1:PRINT TR$;SC;NR$:RE
TURN
2420 REM ::::::::::::::::::::::     [1736]
2430 REM :                    :     [419]
2440 REM :  PAGE PRESENTATION  :    [1858]
2450 REM :                    :     [419]
2460 REM ::::::::::::::::::::::     [1736]
2470 EVERY 5,2 GOSUB 2790:WHILE INK [2448]
EY$<>"":WEND
2480 GOSUB 2690:PLOT -10,-10,1:zc=2 [3408]
00:zd=128:za=150:zb=70:GOSUB 2500
2490 zc=200:zd=124:za=150:zb=70:GOS [3555]
UB 2500:GOTO 2530
2500 ORIGIN zc,zd:zf=2/SQR(za*za+zb [5325]
*zb):zf1=zf*za/zb:zf2=zf*zb/za:zc=z
a:zd=0
2510 zim=1+PI/zf/2:FOR zi=0 TO zim: [5238]
PLOT zc,zd:PLOT zc,-zd:PLOT -zc,zd:
PLOT -zc,-zd
2520 zc=zc-zf1*zd:zd=zd+zf2*zc:NEXT [1895]
 zi:RETURN
2530 ORIGIN 0,0:FOR H=62 TO 194 STE [3251]
P 4:X=200:X1=X
2540 IF TEST(X,H)=1 THEN X=X1:GOTO  [3948]
2550 ELSE PLOT X,H,3:X=X+4:GOTO 254
0
2550 IF TEST(X,H)=1 THEN 2560 ELSE  [3905]
PLOT X,H,3:X=X-4:GOTO 2550
2560 NEXT                           [350]
2570 PEN 12:LOCATE 1,1:PRINT tr$    [2118]
2580 LOCATE 12,17:PRINT STRING$(4,C [4976]
HR$(143)):LOCATE 12,18:PRINT STRING
$(4,CHR$(143))
2590 LOCATE 12,17:PEN 2:PRINT STRIN [5938]
G$ (4,CHR$(206))+CHR$(133):LOCATE 1
2,18:PRINT STRING$ (4,CHR$(206))+CH
R$(133)
2600 PEN 5:LOCATE 3,23:PRINT "CONST [7020]
RUCTION KIT":PEN 4:LOCATE 14,6:PRIN
T "PING":PEN 11:LOCATE 15,8:PRINT "
et"
2610 PEN 6:LOCATE 14,10:PRINT "PONG [3727]
":TAG:PLOT -10,-10,5:MOVE 40,360:PR
INT lmc$;:TAGOFF
2620 mu=REMAIN(2):POKE &A024,&AE:x1 [2780]
=3:x2=8:d1=1:d2=1:ENT 4,15,-1,3:ENV
 4,15,-1,3
2630 CALL &A00D,FN po(4,x1),sp(3):C [3033]
ALL &A00D,FN po(7,x2),sp(4)
2640 FOR t=1 TO 50:NEXT             [997]
2650 CALL &A00D,FN po(4,x1),sp(3):C [3033]
ALL &A00D,FN po(7,x2),sp(4)
2660 x1=x1+d1:IF x1=17 THEN d1=-1:S [4532]
OUND 4,200,45,15,4,4 ELSE IF x1=3 T
HEN d1=1
2670 x2=x2+d2:IF x2=17 THEN d2=-1:S [3099]
OUND 4,200,45,15,4,4 ELSE IF x2=4 T
HEN d2=1
2680 a$=INKEY$:IF a$="" THEN 2630 E [2081]
LSE 2800
2690 FOR H=1 TO 5:MOVE 30,H:DRAW 61 [4236]
0,H,12:NEXT:FOR H=6 TO 20:MOVE 30,H
:DRAW 610,H,6:NEXT
2700 FOR H=20 TO 22:MOVE 30,H:DRAW  [5203]
610,H,1:NEXT:FOR H=1 TO 8:MOVE H,26
:DRAW H,374,12:NEXT
2710 FOR H=8 TO 20:MOVE H,26:DRAW H [4710]
,374,6:NEXT:FOR H=20 TO 22:MOVE H,2
6:DRAW H,374,1:NEXT
2720 FOR H=616 TO 620:MOVE H,26:DRA [6285]
W H,374,1:NEXT:FOR H=620 TO 632:MOV
E H,26:DRAW H,374,6:NEXT
2730 FOR H=632 TO 636:MOVE H,26:DRA [5268]
W H,374,12:NEXT:FOR H=394 TO 400:MO
VE 30,H:DRAW 610,H,12:NEXT
2740 FOR H=382 TO 392:MOVE 30,H:DRA [3677]
W 610,H,6:NEXT:FOR H=378 TO 380:MOV
E 30,H:DRAW 610,H,1:NEXT
2750 X=20:FOR T=1 TO 22:MOVE T,X:DR [3689]
AW T,20,12:X=X-1:NEXT T
2760 X=380:FOR T=1 TO 22:MOVE T,X:D [3296]
RAW T,380:X=X+1:NEXT T
2770 X=380:FOR T=618 TO 636:MOVE T, [3310]
380:DRAW 618,X:X=X+1:NEXT T
2780 X=1:FOR T=618 TO 636:MOVE T,20 [2848]
:DRAW T,X:X=X+1:NEXT T:RETURN
2790 SOUND 1,INT(RND*600)+60,5,15:R [2013]
ETURN
2800 POKE &A024,0:CLS:RETURN        [1048]
2810 REM ::::::::::::::::::::::     [1736]
2820 REM :                    :     [419]
2830 REM :   MENU CARREFOUR   :     [1291]
2840 REM :                    :     [419]
2850 REM ::::::::::::::::::::::     [1736]
2860 WHILE INKEY$<>"":WEND:ga=74:da [4528]
=75:ha=72:ba=73:fe=76
2870 MODE 1:PEN 3:LOCATE 11,5:PRINT [9920]
 STRING$(21,"*"):LOCATE 11,19:PRINT
 STRING$(21,"*"):FOR h=6 TO 18:LOCA
TE 11,h:PRINT "*":LOCATE 31,h:PRINT
"*":NEXT
2880 PEN 2:LOCATE 14,7:PRINT"1 -";: [4453]
PEN 1:PRINT" EXPLICATIONS"
2890 PEN 2:LOCATE 14,10:PRINT"2 -"; [4341]
:PEN 1:PRINT" JOYSTICK"
2900 PEN 2:LOCATE 14,13:PRINT"3 -"; [4118]
:PEN 1:PRINT" CLAVIER"
2910 PEN 2:LOCATE 14,16:PRINT"4 -"; [3177]
:PEN 1:PRINT" CONSTRUCTION"
2920 A$=INKEY$:IF a$="" THEN GOTO 2 [2024]
920
```

```
2930 PRINT CHR$(7):IF A$="1" THEN 3 [2214]
030
2940 IF A$="2" THEN 4000              [607]
2950 IF A$="3" THEN 4010              [1107]
2960 IF A$="4" THEN 3130              [742]
2970 GOTO 2920                        [339]
2980 REM ::::::::::::::::::::::       [1736]
2990 REM :                    :       [419]
3000 REM :    EXPLICATIONS    :       [623]
3010 REM :                    :       [419]
3020 REM ::::::::::::::::::::::       [1736]
3030 CLS:LOCATE 15,1:PRINT"PING & P [5457]
ONG":PEN 2:LOCATE 15,2:PRINT"------
-----"
3040 PEN 3:LOCATE 1,4:PRINT"   Deux [11772]
 petites baballes sont attendues av
ec impatience pour disputer une par
tiede tennis de table."
3050 LOCATE 1,7:PRINT"   Mais avant [8801]
 tout,elles doivent sortir de leur
univers inhospitalier."
3060 LOCATE 1,9:PRINT"   Une clef,p [6041]
our ouvrir deux portes,celadevrait
etre simple."
3070 LOCATE 1,11:PRINT"   Cela l'es [14820]
t moins,lorsque l'on sait   qu'elle
s ne possedent qu'un cerveau pourde
ux,et qu'elles font donc ,ensemble,
les memes gestes."
3080 LOCATE 1,15:PRINT"   Heureusem [14117]
ent,quelques pots de glue , des inv
ersions de commandes  et meme un pe
tit monstre sont la pour faciliter
ou compliquer votre tache."
3090 LOCATE 1,19:PRINT"   Les cas d [9784]
e blocage etant nombreux,la touche
FIRE ou COPY  vous  permettra de re
lancer le tableau."
3100 PEN 1:LOCATE 1,22:PRINT"   Si  [6854]
5 tableaux ne vous suffisent pas ;f
aites les votres..."
3110 LOCATE 33,25:PEN 2:PRINT "<ENT [4596]
ER>":WHILE INKEY$="":WEND:GOTO 2860

3120 REM :::: EXPLICATION KIT ::::   [1164]
3130 CLS:LOCATE 12,1:PRINT"CONSTRUC [3729]
TION KIT":PEN 2:LOCATE 12,2:PRINT"-
----------------"
3140 PEN 3:LOCATE 1,4:PRINT"   La c [8278]
onstruction d'un tableau se fait av
ec les touches flechees,COPY et ENT
ER"
3150 PEN 3:LOCATE 1,8:PRINT"   En h [6461]
aut de l'ecran,tout les elements do
nt vous avez besoin."
3160 LOCATE 1,11:PRINT"   Les 5 pre [8671]
miers sont uniques;une fois en plac
e,ils deviennent inamovibles."
3170 LOCATE 1,14:PRINT"   La touche [11068]
 FIRE/COPY valide l'element choisi
a la place de la mire.La touche  EN
TER remet la mire en haut de l'ecra
n."
3180 LOCATE 1,19:PRINT"    < C > : C [2299]
LS du tableau en cours."
3190 LOCATE 1,21:PRINT"    < S > : S [4886]
AVE du fichier binaire"
3200 LOCATE 1,23:PRINT"    < P > : P [4636]
LAY immediat du tableau "
3210 LOCATE 33,25:PEN 2:PRINT "<ENT [4591]
ER>":WHILE INKEY$="":WEND:MODE 0:GO
TO 3330
3220 REM ::::::: TEMPO ::::::::      [2182]
3230 AFTER 300,3 GOSUB 3250          [794]
3240 SOUND 1,300,20,5,,,15:GOTO 117 [2128]
0
3250 DI:IF pp1=0 THEN pa1=0          [1087]
3260 IF pp2=0 THEN pa2=0             [849]
3270 INK 1,24:EI:RETURN              [1097]
3280 REM ::::::::::::::::::::::      [1736]
3290 REM :                    :      [419]
3300 REM :  CONSTRUCTION KIT  :      [1311]
3310 REM :                    :      [419]
3320 REM ::::::::::::::::::::::      [1736]
3330 CALL &A039,&C050,SP(2):FOR h=0 [2955]
 TO 208:POKE tabl(6)+h,1:NEXT
3340 FOR h=0 TO 40:POKE tabl(7)+h,0 [2341]
:NEXT
3350 LOCATE 1,1:PRINT STRING$(20,"  [3016]
"):LOCATE 1,2:PRINT STRING$(20," ")
3360 RESTORE 3390:FOR h=0 TO 76 STE [1954]
P 4:READ a
3370 IF a<16 THEN CALL &A00D,&C000+ [2889]
h,sp(a)
3380 NEXT                            [350]
3390 DATA 3,5,9,10,9,7,11,1,2,15,12 [2328]
,14,13,15,21,15,22,15,23,15
3400 RESTORE 3410:FOR h=0 TO 19:REA [2973]
D a:POKE tabl(7)+h,a:NEXT
3410 DATA 3,4,6,7,6,5,8,1,2,0,9,11, [2586]
10,0,12,0,13,0,14,0
3420 FOR h=1 TO 13:FOR g=1 TO 21:je [1939]
u(g,h)=2:NEXT g,h
3430 FOR h=2 TO 12:FOR g=2 TO 20:je [1982]
u(g,h)=1:NEXT g,h
3440 DATA 3,4,6,7,6,5,8,1,2,0,9,11, [2586]
10,0,12,0,13,0,14,0
3450 TAG:PLOT -10,-10,6:MOVE 448,39 [4142]
2:PRINT "C S P";:TAGOFF
3460 POKE &A024,&AE:CALL &A00D,&C00 [1650]
0,sp(16):cx=0
3470 a$=INKEY$:IF a$="" THEN 3470   [1395]
3480 IF a$=CHR$(243) THEN 3510      [461]
3490 IF a$=CHR$(242) THEN 3530      [881]
3500 IF a$=CHR$(224) THEN 3550 ELSE [1436]
 3470
3510 CALL &A00D,&C000+cx,sp(16):cx= [2428]
cx+4:IF cx>76 THEN cx=0
3520 CALL &A00D,&C000+cx,sp(16):GOT [1825]
O 3470
3530 CALL &A00D,&C000+cx,sp(16):cx= [3335]
cx-4:IF cx<0 THEN cx=76
3540 CALL &A00D,&C000+cx,sp(16):GOT [1825]
O 3470
3550 sx=(cx/4)+1:RESTORE 3390:FOR h [2631]
=1 TO sx:READ a:NEXT
3560 IF a>20 THEN 3830              [527]
3570 IF PEEK(tabl(7)+(sx-1))=0 THEN [2275]
 a=15
3580 x=10:y=13:x1=x:y1=y:add=104:CA [4089]
LL &A00D,FN po(x,y),sp(16)
3590 k$=INKEY$:IF k$="" THEN 3590   [1405]
3600 IF k$=CHR$(240) AND y>4 THEN y [1943]
=y-2:add=add-19:GOTO 3670
3610 IF k$=CHR$(241) AND y<22 THEN  [3292]
y=y+2:add=add+19:GOTO 3670
3620 IF k$=CHR$(242) AND x>1 THEN x [1078]
=x-1:add=add-1:GOTO 3670
```

```
3630 IF k$=CHR$(243) AND x<19 THEN  [2723]
x=x+1:add=add+1:GOTO 3670
3640 IF k$=CHR$(224) THEN 3700      [1156]
3650 IF k$=CHR$(13) THEN 3740       [433]
3660 GOTO 3590                      [491]
3670 CALL &A00D,FN po(x1,y1),sp(16) [1753]
3680 CALL &A00D,FN po(x,y),sp(16):x [2045]
1=x:y1=y:GOTO 3590
3690 REM ::: AFF SPRITE CHOISI :::  [1522]
3700 c=PEEK(tabl(6)+add):IF c>2 AND [4148]
 c<8 THEN PRINT CHR$(7):GOTO 3590
3710 POKE &A024,0:CALL &A00D,FN po( [2155]
x,y),sp(a)
3720 b=PEEK(tabl(7)+(sx-1)):POKE ta [5699]
bl(6)+add,b:IF b>2 AND b<8 THEN 375
0
3730 POKE &A024,&AE:CALL &A00D,FN p [3062]
o(x,y),sp(16):GOTO 3590
3740 CALL &A00D,FN po(x,y),sp(16):C [3635]
ALL &A00D,&C000+cx,sp(16):GOTO 3460
3750 CALL &A00D,&C000+cx,sp(15):CAL [6082]
L &A00D,&C000+cx,sp(16):code=tabl(7
)+(sx-1):POKE code,0:POKE &A024,&AE
3760 IF b=3 THEN POKE tabl(7)+20,x: [3858]
POKE tabl(7)+21,y:GOTO 3470
3770 IF b=4 THEN POKE tabl(7)+22,x: [3809]
POKE tabl(7)+23,y:GOTO 3470
3780 IF b=6 AND PEEK(tabl(7)+24)<>0 [4947]
 THEN POKE tabl(7)+26,x:POKE tabl(7
)+27,y:GOTO 3470
3790 IF b=6  THEN POKE tabl(7)+24,x [2050]
:POKE tabl(7)+25,y:GOTO 3470
3800 IF b=7 THEN 3470               [456]
3810 IF b=5 THEN POKE tabl(7)+28,1: [5511]
POKE tabl(7)+29,x:POKE tabl(7)+30,y
:GOTO 3470
3820 GOTO 3470                      [475]
3830 b=a-20:ON b GOTO 3850,3880,392 [2722]
0
3840 REM ::::      CLS      ::::    [595]
3850 PEN 6:LOCATE 5,25:PRINT "CLS   [3467]
":GOSUB 3950
3860 IF rep=1 THEN POKE &A024,0:GOT [3139]
O 3330 ELSE 3470
3870 REM ::::      SAVE     ::::    [770]
3880 PEN 6:LOCATE 5,25:PRINT "SAVE  [2363]
":GOSUB 3950
3890 IF rep=1 THEN 3900 ELSE 3470   [1563]
3900 SAVE "!ping",b,&96EC,&AA0:GOTO [3519]
 3470
3910 REM ::::      PLAY     ::::    [711]
3920 PEN 6:LOCATE 5,25:PRINT "PLAY  [2996]
":GOSUB 3950
3930 IF rep=1 THEN POKE tabl(7)+31, [3614]
1:ta=6:POKE &A024,0:CLS:GOTO 380
3940 GOTO 3470                      [475]
3950 PEN 6:LOCATE 10,25:PRINT "? (Y [2071]
/N)":rep=0
3960 a$=INKEY$:IF a$="" THEN 3960   [1352]

3970 a$=UPPER$(a$):IF a$="Y" THEN r [2285]
ep=1:GOTO 3990
3980 IF a$="N" THEN 3990 ELSE 3960  [1078]

3990 LOCATE 5,25:PRINT STRING$(12," [3811]
 "):RETURN
4000 ga=74:da=75:ha=72:ba=73:fe=76: [3185]
MODE 0:GOTO 380
4010 ga=8:da=1:ha=0:ba=2:fe=9:MODE  [2537]
0:GOTO 380
```

```
96EC:00 08 04 04 0C 00 08 04:AA
96F4:10 04 04 08 0C 08 04 24:E6
96FC:08 24 18 00 10 08 04 04:F6
9704:24 04 04 0C 08 24 08 08:0F
970C:04 24 24 04 08 0C 04 08:13
9714:0C 0C 18 0C 08 18 08 00:0F
971C:0C 04 04 0C 08 24 10 24:33
9724:04 08 0C 0C 08 0C 08 08:03
972C:44 CC 44 CC 10 CC 10 CC:9B
9734:10 30 10 30 00 00 00 00:4B
973C:CC 44 CC 44 CC 10 CC 10:AB
9744:30 10 30 10 00 00 00 00:5B
974C:44 CC 44 CC 10 CC 10 CC:BB
9754:10 30 10 30 00 00 00 00:6B
975C:CC 44 CC 44 CC 10 CC 10:CB
9764:30 10 30 10 00 00 00 00:7B
976C:00 40 80 00 00 C0 C0 00:43
9774:40 C0 C0 80 40 C0 C0 80:8B
977C:C0 80 80 C0 C0 20 80 60:53
9784:C0 94 94 C0 C0 34 94 60:AB
978C:C0 90 90 C0 C4 C0 C0 C0:C7
9794:C4 C8 C0 C8 C0 CC CC C8:5F
979C:40 C4 CC 80 40 C0 C8 80:CB
97A4:00 C0 C0 00 00 40 80 00:7B
97AC:00 40 80 00 00 C0 C0 00:83
97B4:40 C0 C0 80 40 C0 C0 80:CB
97BC:C0 40 40 C0 90 40 10 C0:F3
97C4:C0 68 68 C0 90 68 38 C0:9B
97CC:C0 60 60 C0 C0 C0 C0 C8:AB
97D4:C4 C0 C4 C8 C4 CC CC C0:97
97DC:40 CC C8 80 40 C4 C0 80:0B
97E4:00 C0 C0 00 00 40 80 00:BB
97EC:00 00 00 00 00 40 80 00:43
97F4:00 C0 C0 00 40 C0 C0 80:4B
97FC:40 C0 C0 80 C0 90 90 C0:73
9804:C0 34 94 60 C0 34 94 60:6C
980C:C0 90 90 C0 C0 C0 C0 C0:44
9814:C4 C0 C0 C8 40 CC CC 80:10
981C:40 C4 C8 80 00 C0 C0 00:80
9824:00 40 80 00 00 00 00 00:7C
982C:00 00 00 00 00 40 80 00:84
9834:00 C0 C0 00 40 C0 C0 80:8C
983C:40 C0 C0 80 C0 60 60 C0:54
9844:90 68 38 C0 90 68 38 C0:BC
984C:C0 60 60 C0 C0 C0 C0 C0:24
9854:C4 C0 C0 C8 40 CC CC 80:50
985C:40 C4 C8 80 00 C0 C0 00:C0
9864:00 40 80 00 00 00 00 00:BC
986C:10 20 10 20 30 10 20 30:F4
9874:20 20 10 10 20 00 00 10:9C
987C:10 14 20 20 00 3C 3C 00:F0
9884:50 14 28 A0 70 A0 50 B0:58
988C:30 F0 F0 30 20 F0 F0 10:74
9894:00 E4 D8 00 00 04 08 00:F4
989C:00 4C 8C 00 00 04 08 00:18
98A4:00 64 98 00 00 0C 0C 00:50
98AC:00 10 30 00 00 20 00 20:C4
98B4:10 00 00 B8 20 00 00 B8:EC
98BC:20 00 00 B2 20 00 00 B2:F8
98C4:20 00 00 B8 20 00 00 B8:0C
98CC:20 00 00 B8 20 00 00 B8:14
98D4:20 00 00 B8 20 00 00 B8:1C
98DC:20 00 00 B8 20 00 00 B2:1E
98E4:20 00 00 B2 30 30 30 30:0E
98EC:00 44 0C 00 00 D9 F9 08:AE
98F4:44 F9 F9 AC DC F9 F9 F9:35
98FC:F6 F9 AC 59 F6 F9 0C 0C:8F
9904:F6 F9 F9 F9 F6 F9 F9 F9:5F
990C:F6 F9 F9 F9 C0 F9 F9 F9:31
9914:E2 F9 F9 F9 F6 F9 F9 F9:5B
991C:F6 F9 F9 F9 F6 F9 AC 59:8A
9924:F6 F9 0C 0C F3 F9 F9 F9:A2
992C:00 30 80 00 10 40 40 00:05
9934:10 40 40 00 10 C0 C0 00:ED
993C:40 40 40 00 40 40 40 00:55
9944:00 C0 80 00 00 40 00 00:5D
994C:00 40 00 00 00 40 00 00:65
9954:00 40 00 20 00 40 90 00:1D
995C:00 40 80 80 00 40 40 80:35
9964:00 40 90 00 00 40 00 20:2D
996C:54 FC FC 00 AC 0C 0C A8:BD
9974:56 0C 5C A2 47 FC F9 02:AB
997C:23 CF 8B A2 67 03 53 A2:93
9984:67 CF DB 8A 67 C3 D3 8A:3F
998C:67 C7 CF 8A 67 C7 CF 22:CB
9994:67 C7 C7 22 33 C7 C7 22:27
999C:33 C3 C7 22 33 CF 9B 22:D3
99A4:33 CF 9B 00 11 33 22 00:40
99AC:00 00 00 00 00 00 00 00:45
99B4:00 00 00 00 F0 00 00 20:5D
99BC:3C 3C 3C 30 F0 00 00 20:49
99C4:00 00 00 00 00 00 00 00:5D
99CC:00 00 00 00 00 00 00 00:65
99D4:10 00 00 F0 30 3C 3C 3C:51
99DC:10 00 00 F0 00 00 00 00:75
99E4:00 00 00 00 00 00 00 00:7D
99EC:B4 A0 00 20 B4 A0 10 30:8D
99F4:B4 A0 00 28 B4 A0 00 28:85
99FC:14 00 00 28 14 00 00 28:0D
9A04:14 00 00 28 14 00 00 28:16
9A0C:14 00 00 28 14 00 00 28:1E
9A14:14 00 00 28 14 00 00 28:26
9A1C:14 00 50 78 14 00 50 78:6E
9A24:30 20 50 78 10 00 50 78:AE
9A2C:00 00 00 00 01 02 00 00:C9
9A34:07 02 00 00 01 02 00 01:DB
9A3C:00 02 00 45 00 03 00 16:36
```

```
9A44:00 03 03 47 00 47 29 9E:39
9A4C:00 47 03 28 00 01 CF 00:28
9A54:10 00 02 10 00 28 02 28:62
9A5C:00 14 16 00 00 14 3C 00:70
9A64:00 28 82 28 10 00 82 10:72
9A6C:00 00 00 00 00 00 00 00:06
9A74:00 00 00 00 00 00 00 00:0E
9A7C:00 00 00 00 00 00 00 00:16
9A84:00 00 00 00 00 00 00 00:1E
9A8C:00 00 00 00 00 00 00 00:26
9A94:00 00 00 00 00 00 00 00:2E
9A9C:00 00 00 00 00 00 00 00:36
9AA4:00 00 00 00 00 00 00 00:3E
9AAC:33 33 33 33 22 00 00 11:45
9AB4:22 00 00 11 22 00 00 11:B4
9ABC:22 00 00 11 22 00 00 11:BC
9AC4:22 00 00 11 22 00 00 11:C4
9ACC:22 00 00 11 22 00 00 11:CC
9AD4:22 00 00 11 22 00 00 11:D4
9ADC:22 00 00 11 22 00 00 11:DC
9AE4:22 00 00 11 33 33 33 33:7D
9AEC:01 01 01 01 01 01 01 01:8E
9AF4:01 01 01 08 01 01 01 01:9D
9AFC:01 01 01 01 01 01 01 00:9D
9B04:00 00 01 01 01 01 01 01:A5
9B0C:01 01 01 01 01 07 01 01:B5
9B14:01 01 00 00 00 02 01 01:B5
9B1C:02 02 02 02 02 02 02 02:C7
9B24:02 01 01 01 01 00 00 03:C8
9B2C:02 01 01 01 01 01 01 01:D0
9B34:01 01 01 01 01 01 01 01:D7
9B3C:01 01 01 02 01 01 01 01:E0
9B44:01 08 01 01 01 01 01 02:EF
9B4C:02 02 02 02 02 02 02 01:F6
9B54:01 01 01 01 01 01 01 01:F7
9B5C:01 01 01 01 01 01 01 01:FF
9B64:01 02 02 02 02 02 02 02:0E
9B6C:02 02 01 01 01 01 01 01:11
9B74:01 01 01 01 02 00 00 00:15
9B7C:01 01 01 01 01 01 01 01:1F
9B84:01 01 01 01 01 01 01 01:27
9B8C:00 00 00 01 01 01 01 01:2C
9B94:01 01 01 01 01 01 01 01:37
9B9C:01 01 01 00 00 04 01 01:40
9BA4:01 01 01 01 01 01 06 01:4C
9BAC:01 01 01 01 01 01 01 01:4F
9BB4:01 01 01 01 01 01 01 01:57
9BBC:06 06 01 01 01 01 01 09:71
9BC4:01 01 02 01 01 01 01 01:68
9BCC:01 01 01 01 02 02 01 01:71
9BD4:01 01 01 01 01 02 01 01:78
9BDC:01 01 02 01 08 01 07 01:8D
9BE4:01 01 01 02 02 01 01 01:89
9BEC:02 01 01 01 01 02 01 02:92
9BF4:02 02 01 01 01 01 02 01:9A
9BFC:01 01 01 01 01 01 01 01:9F
9C04:01 01 01 01 01 01 01 01:A8
9C0C:01 02 01 01 0B 01 01 01:BB
9C14:01 02 01 01 01 01 01 01:B9
9C1C:01 01 02 01 01 01 01 01:C1
9C24:01 01 01 01 01 01 01 01:C8
9C2C:01 01 01 01 01 01 01 01:D0
9C34:01 02 01 01 01 01 02 08:E1
9C3C:02 01 01 01 01 01 01 01:E1
9C44:0A 01 01 02 01 01 01 01:F2
9C4C:01 01 02 01 01 01 0B 02:FC
9C54:01 01 02 02 01 01 01 01:FA
9C5C:01 01 01 01 01 01 01 01:00
9C64:01 02 01 01 01 01 01 01:09
9C6C:01 00 00 02 01 01 02 02:11
9C74:01 00 00 01 01 01 02 01:17
9C7C:01 01 01 02 00 03 02 01:23
9C84:01 01 01 01 00 04 01 01:2A
9C8C:01 06 01 01 01 01 01 07:3B
9C94:01 01 01 01 01 01 01 01:38
9C9C:01 01 01 01 01 01 08 05:4B
9CA4:00 00 00 00 00 00 02 01:43
9CAC:03 01 01 01 01 01 01 06:57
9CB4:01 01 00 02 02 02 02 02:5C
9CBC:00 02 01 01 01 01 01 02:61
9CC4:02 02 02 01 01 00 01 01:6A
9CCC:01 01 01 00 02 01 01 01:70
9CD4:01 01 01 01 01 01 01 01:78
9CDC:00 01 01 01 01 01 00 02:7F
9CE4:01 01 01 01 01 01 01 01:88
9CEC:01 01 01 00 00 00 00 01:8C
9CF4:01 00 02 04 01 01 00 00:99
9CFC:00 01 01 08 01 01 01 01:A6
9D04:01 00 01 01 00 00 00 00:A4
9D0C:00 00 02 00 01 01 01 01:AF
9D14:01 01 01 01 00 01 01 01:B8
9D1C:01 01 01 01 01 02 00 01:C1
9D24:01 01 01 01 01 01 02 00:C9
9D2C:02 02 02 02 01 01 01 01:D5
9D34:02 00 01 01 01 01 01 01:D9
9D3C:01 02 00 00 00 00 00 00:DC
9D44:00 00 00 00 00 01 01 01:E4
9D4C:01 01 01 06 02 01 01 01:F7
9D54:01 01 01 01 01 01 01 02:FA
9D5C:01 01 01 05 00 00 00 00:01
9D64:00 00 00 00 00 00 00 00:01
9D6C:00 00 00 0A 02 06 01 02:1E
9D74:01 01 06 02 01 01 01 01:1F
9D7C:01 01 01 01 01 01 01 02:22
9D84:01 01 02 02 02 02 02 01:2E
9D8C:01 01 01 01 01 01 01 01:31
9D94:01 01 02 01 01 01 01 01:3A
9D9C:01 01 01 01 01 01 01 01:41
9DA4:02 02 02 02 01 02 01 02:4F
9DAC:02 02 02 01 01 01 01 01:54
9DB4:01 01 01 01 01 01 01 01:59
9DBC:02 01 02 01 01 01 01 01:63
9DC4:01 02 01 01 02 01 01 01:6B
9DCC:01 01 03 02 01 02 01 02:76
9DD4:01 01 01 01 02 01 01 02:7B
9DDC:01 01 01 01 01 01 02 01:82
9DE4:08 01 02 01 01 01 01 02:92
9DEC:01 01 02 01 01 01 01 01:92
9DF4:01 02 01 02 01 02 01 01:9C
9DFC:02 01 01 01 01 02 04 01:A6
9E04:01 01 01 01 02 01 02 01:AC
9E0C:02 02 02 02 01 02 02 02:B9
9E14:02 02 02 02 02 02 02 02:C2
9E1C:01 07 0A 02 01 01 01 01:D2
9E24:01 01 01 01 01 01 01 01:CA
9E2C:01 01 01 01 01 01 01 01:D2
9E34:01 09 01 01 01 01 01 01:E2
9E3C:0A 01 01 01 01 01 01 01:EB
9E44:01 05 00 00 00 00 00 02:EA
9E4C:01 01 01 01 01 01 01 01:F2
9E54:01 06 01 01 00 01 01 01:FE
9E5C:01 00 02 01 01 01 01 02:03
9E64:02 02 02 02 02 01 01 00:0E
9E6C:01 01 03 01 00 02 0B 01:1E
9E74:01 01 01 01 01 07 0B 01:2A
9E7C:01 09 00 01 01 01 01 00:28
9E84:02 02 02 02 01 01 09 02:37
9E8C:02 02 01 01 01 00 00 00:31
9E94:00 00 00 00 00 00 00 00:32
9E9C:00 00 00 00 02 01 02 08:47
9EA4:02 01 02 02 02 00 01 01:4D
9EAC:01 01 00 01 01 01 00 02:51
9EB4:01 02 02 02 01 02 01 01:5E
9EBC:00 01 01 04 01 02 01 0A:6E
9EC4:01 00 02 01 02 06 02 01:71
9ECC:02 0B 01 00 01 01 01 01:7C
9ED4:02 01 01 00 00 02 01 02:7C
9EDC:01 02 01 02 02 02 00 00:84
9EE4:00 00 00 00 00 00 00 00:82
9EEC:02 01 01 01 01 01 01 01:93
9EF4:01 01 01 01 0A 01 01 01:A3
9EFC:01 01 01 01 01 00 00 00:9F
9F04:00 00 00 00 00 00 00 00:A3
9F0C:00 00 00 00 00 00 00 00:AB
9F14:00 00 00 00 00 00 00 00:B3
9F1C:00 00 00 00 00 00 00 00:BB
9F24:00 00 00 00 00 00 00 00:C3
9F2C:00 00 00 00 00 00 00 00:CB
9F34:00 00 00 00 00 00 00 00:D3
9F3C:00 00 00 00 00 00 00 00:DB
9F44:00 00 00 00 00 00 00 00:E3
9F4C:00 00 00 00 00 00 00 00:EB
9F54:00 00 00 00 00 00 00 00:F3
9F5C:00 00 00 00 00 00 00 00:FB
9F64:00 00 00 00 00 00 00 00:03
9F6C:00 00 00 00 00 00 00 00:0B
9F74:00 00 00 00 00 00 00 00:13
9F7C:00 00 00 00 00 00 00 00:1B
9F84:00 00 00 00 00 00 00 00:23
9F8C:00 00 00 00 00 00 00 00:2B
9F94:00 00 00 00 00 00 00 00:33
9F9C:00 00 00 00 00 00 00 00:3B
9FA4:00 00 00 00 00 00 00 00:43
9FAC:00 00 00 00 00 00 00 00:4B
9FB4:00 00 00 00 00 00 00 00:53
9FBC:00 00 00 00 00 00 00 00:5B
9FC4:00 00 00 00 00 00 00 00:63
9FCC:00 00 00 00 00 00 00 00:6B
9FD4:00 00 00 00 00 00 00 00:73
9FDC:00 00 00 00 00 00 00 00:7B
9FE4:00 00 00 00 00 00 00 00:83
9FEC:00 00 00 00 00 00 00 00:8B
9FF4:00 00 00 00 00 00 00 00:93
9FFC:00 00 00 00 7C C6 08 67:4C
A004:D0 D5 11 50 C0 19 D1 C9:1D
A00C:00 DD 5E 00 DD 56 01 DD:F8
A014:6E 02 DD 66 03 06 10 CD:4D
A01C:19 BD F3 C5 E5 06 04 1A:53
A024:00 77 13 23 10 F9 E1 CD:28
A02C:00 A0 C1 10 EE FB C9 00:EF
A034:00 90 14 82 C7 DD 5E 00:FC
A03C:DD 56 01 ED 53 34 A0 DD:01
A044:6E 02 DD 66 03 22 37 A0:93
A04C:CD 72 A0 21 A2 C0 22 37:A7
A054:A0 2A 34 A0 01 40 00 ED:C0
A05C:42 22 34 A0 3A 36 A0 3D:81
A064:32 36 A0 06 0B CD 74 A0:FE
A06C:3E 14 32 36 A0 C9 06 0C:41
A074:C5 2A 37 A0 3A 36 A0 47:31
A07C:C5 E5 ED 5B 34 A0 06 10:F8
A084:CD 1F A0 E1 01 04 00 09:9F
A08C:C1 10 ED 2A 37 A0 01 A0:8C
A094:00 09 22 37 A0 C1 10 D8:DF
A09C:C9 00 00 00 82 C7 82 C7:97
A0A4:D1 95 00 90 21 A2 C0 22:DF
A0AC:A0 A0 22 A2 A0 DD 6E 00:3B
A0B4:DD 66 01 22 A6 A0 DD 6E:4B
A0BC:02 DD 66 03 22 A4 A0 06:10
A0C4:0B 2A A0 A0 C5 E5 06 13:9C
A0CC:2A A2 A0 C5 E5 2A A4 A0:F0
A0D4:7E 23 22 A4 A0 2A A6 A0:EB
A0DC:CD FE A0 E1 01 04 00 ED:BA
A0E4:4A 22 A2 A0 C1 10 E4 E1:C8
A0EC:01 A0 00 ED 4A 22 A0 A0:C6
A0F4:22 A2 A0 C1 10 CE C9 00:60
A0FC:00 00 FE 00 20 06 01 80:41
A104:03 C3 6F A1 FE 01 20 06:A0
A10C:01 00 00 C3 6F A1 FE 02:81
A114:20 06 01 40 00 C3 6F A1:EF
A11C:FE 03 20 06 01 80 00 C3:28
A124:6F A1 FE 04 20 06 01 00:FE
A12C:01 C3 6F A1 FE 05 20 06:CA
A134:01 80 01 C3 6F A1 FE 06:2E
A13C:20 06 01 00 02 C3 6F A1:D9
A144:FE 07 20 06 01 40 02 C3:16
A14C:6F A1 FE 08 20 06 01 80:AA
A154:02 C3 6F A1 FE 09 20 06:F7
A15C:01 C0 02 C3 6F A1 FE 0A:9B
A164:20 06 01 00 03 C3 6F A1:02
A16C:01 40 03 ED 4A E5 D1 2A:68
A174:A2 A0 06 10 C3 1F A0 00:EF
A17C:00 00 00 00 00 00 00 00:1D
A184:00 00 00 00 00 00 00 00:25
A18C:00 00 00 00 00 00 00 00:2D
```



SPRITER & IMPLANT

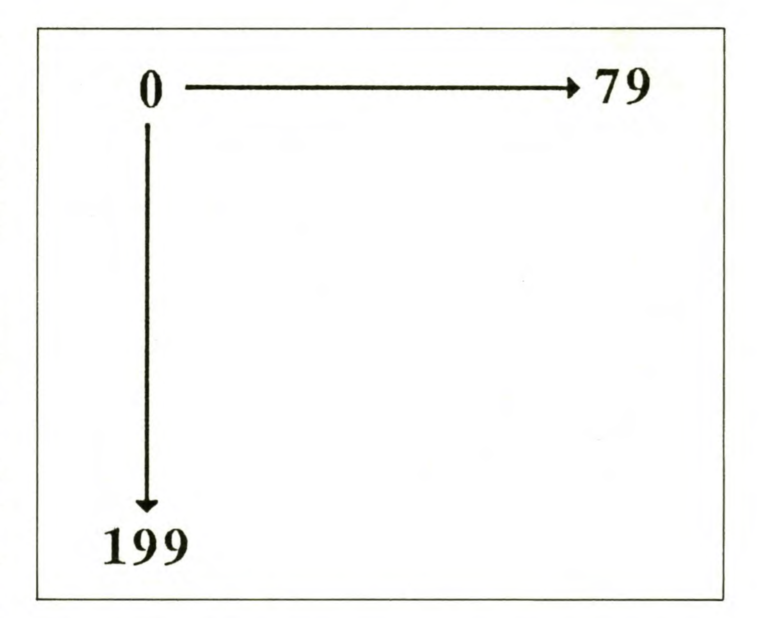
Les éditeurs de sprites classiques ont deux défauts :
— les options qu'ils proposent pour la création des sprites, sont bien souvent insuffisantes (absence d'option de remplissage de zones, de rotation, de zoom, etc.).
— Ils occupent de la place en mémoire et réduisent d'autant le nombre de sprites.
D'où l'originalité de *Spriter*, permettant la création de sprites à partir de pages écrans préalablement réalisées à l'aide d'un utilitaire de DAO quelconque.

Utilisation

O Dessinez des sprites grâce à votre utilitaire de dessin favori et sauvez-les en pages écrans normales.
O Chargez *Spriter*. Modifiez si nécessaire l'adresse d'implantation des sprites, qui doit être comprise entre 10 000 et 42 000 afin de ne pas empiéter sur les RSX (cf. programme *Implant*).
O Sélectionnez l'option **Chargement écran**. Déplacez au moyen du joystick, le rectangle qui figure sur l'écran et modifiez sa taille par les touches fléchées. Validez par FIRE ou COPY ; le sprite est alors codé. ENTER permet le retour au menu principal, ce qui est très utile pour charger une page écran différente et y saisir d'autres sprites. Votre fichier sprites au complet, sauvez-le par l'option **Sauvegarde sprites**. Le programme précise alors le nombre N de sprites codés et l'adresse de la table des sprites que vous devez noter.
O Après un RESET de l'ordinateur, chargez le programme *Implant* (ou incorporez-le dans vos programmes). Dans tous les cas, faites-le tourner *après* un SYMBOL AFTER. Chargez également votre fichier sprites, après avoir abaissé HIMEM (faites MEMORY imp-1, où « imp » est l'adresse d'implantation de vos sprites).
Vous disposez à présent de deux nouvelles instructions :
— ITABLE,adr : où « adr » désigne l'adresse de la table des sprites que vous avez précédemment notée. Cette instruction ne sert en principe qu'une fois.
— ISPRITE,n,x,y : où « n » précise le numéro du sprite à afficher (de 0 à N-1), « x » son abscisse et « y » son ordonnée.

Quelques précisions

O Les coordonnées sont en octets, la structure de l'écran est donc la suivante :



O Le sprite ne s'affichera pas si n, x ou y sont faux, ou bien encore si le sprite sort de l'écran en raison de sa largeur ou longueur trop excessive.

O L'affichage des sprites se fait en mode XOR. Donc, pour effacer un sprite, il suffit de l'afficher une seconde fois au même endroit.

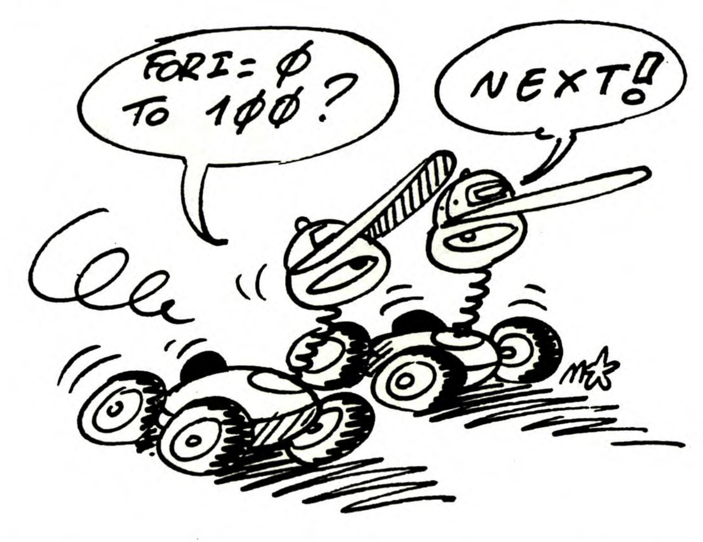
O Dans *Spriter*, la fusion des fichiers de sprites se réalise par l'option **Chargement sprites**, sans modifier l'adresse d'implantation.

Adaptation 464 cassette

Ligne 260 : supprimer le CAT.
Ligne 300 : LOAD "!" + n$,&C000
Lignes 760 et 920 : supprimer PEN 1 : a$="*.spr" : IDIR,@a$
Lignes 780 et 950 : n$="!" + LEFT$ (n$,8) + ".spr"

Hervé Inisian

```
10 ' ****************************** [1252]
20 ' **                          ** [104]
30 ' ** SPRITER ... INISAN HERVE ** [2361]
40 ' **                          ** [104]
50 ' ****************************** [1252]
60 '                                [117]
70 MODE 1                           [506]
80 DATA 0,13,25,26,2,3,4,5,6,7,8,9, [2443]
10,11,12,13
90 READ c:INK 0,c:BORDER c          [851]
100 FOR t=1 TO 15:READ c:INK t,c:NE [1764]
XT
110 OPENOUT"d":MEMORY 9849:CLOSEOUT [2862]
120 '                               [117]
130 ' *** IMPLANTATION LM : TRANSFE [3439]
RT ZONE ECRAN GRAPHIQUE
140 '                               [117]
150 DATA 01,00,00,11,00,00,c5,21,40 [9018]
,01,c1,c5,d5,09,eb,e5,c1,21,c8,00,e
d,42,d5,e5,cd,f0,bb,cd,de,bb,e1,d1,
1b,00,00,00,00,00,cd,ea,bb,d1,13,13
,21,50,00,ed,52,30,d4,c1,03,03,03,0
3,21,7c,00,ed,42,30,c4,c9
160 FOR t=0 TO 63:READ a$:POKE 9850 [3156]
+t,VAL("&"+a$):NEXT
170 '                               [117]
180 ' *** IMPLANTATION LM : CODAGE  [2850]
SPRITE
190 '                               [117]
200 DATA 06,14,21,04,f1,11,00,80,c5 [6227]
,e5,01,0a,00,ed,b0,e1,01,00,08,09,3
0,04,01,50,c0,09,c1,10,eb,ed,53,0a,
27,c9
210 FOR t=0 TO 33:READ a$:POKE 9960 [2701]
+t,VAL("&"+a$):NEXT
220 '                               [117]
230 ' *** INITIALISATION            [1290]
240 '                               [117]
250 adimpl=10000:depart=adimpl:GOTO [1834]
 980
260 MODE 1:CAT:PEN 2:PRINT"Nom de l [8706]
'ecran a charger (+extension) :":IN
PUT"(ENTER pour abandonner) : ",n$:
IF n$=""GOTO 980
270 PRINT:INPUT"Mode ecran (0/1) :  [2201]
",md:IF md<0 OR md>1 GOTO 270
280 IF md=1 THEN dpl=2:pok=0 ELSE d [2592]
pl=4:pok=3
290 POKE 9904,pok:POKE 9905,pok     [1108]
300 MODE md:LOAD n$,&C000:x=320:y=2 [3705]
00:lar=20:lon=20
310 '                               [117]
320 ' *** NOYAU                     [451]
330 '                               [117]
340 PRINT CHR$(23)CHR$(1):adl=adimp [2287]
l
350 MOVE x,y:DRAWR lar,0,1:DRAWR 0, [3438]
-lon:DRAWR -lar,0:DRAWR 0,lon
360 a$=LOWER$(INKEY$):a=JOY(0)      [2195]
370 IF a$=""AND a=0 GOTO 360        [1446]
380 MOVE x,y:DRAWR lar,0,1:DRAWR 0, [3438]
-lon:DRAWR -lar,0:DRAWR 0,lon
390 IF a<>0 GOTO 400 ELSE 450       [2278]
400 IF a=1 THEN IF y<397 THEN y=y+2 [3036]
:GOTO 350
410 IF a=2 THEN IF y-lon>1 THEN y=y [2346]
-2:GOTO 350
420 IF a=4 THEN IF x>dpl-1 THEN x=x [1616]
-dpl:GOTO 350
430 IF a=8 THEN IF x+lar<639-dpl TH [3215]
EN x=x+dpl:GOTO 350
440 IF a=16 GOTO 540 ELSE 350       [1818]
450 IF a$=CHR$(240)THEN IF lon>1 TH [2552]
EN lon=lon-2:GOTO 350
460 IF a$=CHR$(241)THEN IF y-lon>1  [1401]
THEN lon=lon+2:GOTO 350
470 IF a$=CHR$(242)THEN IF lar>dpl- [2955]
1 THEN lar=lar-dpl:GOTO 350
480 IF a$=CHR$(243)THEN IF x+lar<63 [2376]
9-dpl THEN lar=lar+dpl:GOTO 350
490 IF a$=CHR$(13)GOTO 980          [797]
500 IF a$<>CHR$(224)GOTO 350        [1906]
510 '                               [117]
520 ' *** TRANSFERT ZONE ECRAN      [1860]
530 '                               [117]
540 LOCATE 1,1:PRINT CHR$(23)CHR$(0 [968]
)
```




```
550 IF md=0 THEN mlt=x/4 MOD 2 ELSE [3295]
 mlt=x/2 MOD 4
560 IF mlt=0 GOTO 660               [882]
570 POKE 9882,&1B:FOR t=0 TO 4:POKE [1438]
 9883+t,0:NEXT
580 IF md=1 THEN FOR t=0 TO mlt*2-1 [2543]
:POKE 9882+t,&1B:NEXT
590 msb=INT(x/256):lsb=x-msb*256:PO [3284]
KE 9858,lsb:POKE 9859,msb
600 msb=INT(y/256):lsb=y-msb*256:PO [2356]
KE 9868,lsb:POKE 9869,msb
610 msb=INT(lon/256):lsb=lon-msb*25 [4920]
6:POKE 9895,lsb:POKE 9896,msb
620 msb=INT((lar+(mlt+1)*dpl)/256): [5460]
lsb=lar+(mlt+1)*dpl-msb*256:POKE 99
07,lsb:POKE 9908,msb
630 CALL 9850:x=x-dpl*mlt           [2104]
640 '                               [117]
650 ' *** CODAGE                    [582]
660 '                               [117]
670 POKE adimpl,INT(lar/8)+1:POKE a [2690]
dimpl+1,lon/2+1:adimpl=adimpl+2
680 ox=INT(x/8):oy=199-INT(y/2):adr [3619]
1=49152+80*INT(oy/8)+2048*(oy MOD 8
)+ox
690 msb=INT(adr1/256):lsb=adr1-msb* [2713]
256:POKE 9963,lsb:POKE 9964,msb
700 msb=INT(adimpl/256):lsb=adimpl- [2924]
msb*256:POKE 9966,lsb:POKE 9967,msb
710 POKE 9961,INT(lon/2)+1:POKE 997 [2713]
1,INT(lar/8)+1
```

```
720 CALL 9960:adimpl=PEEK(&270A)+25 [2668]
6*PEEK(&270B):GOTO 340
730 '                               [117]
740 ' *** SAUVEGARDE FICHIER        [1684]
750 '                               [117]
760 MODE 1:PEN 2:PRINT"SAUVEGARDE F [13277]
ICHIER":PRINT:PEN 1:a$="*.spr":|DIR
,@a$:PEN 2:PRINT"Nom du fichier san
s extension :":INPUT"(ENTER pour ab
andonner) : ",n$
770 IF n$=""GOTO 980                [1067]
780 GOSUB 820:n$=LEFT$(n$,8)+".spr" [11866]
:PRINT:PEN 3:PRINT"Adresse table :
&"HEX$(adtab)"="adtab:PRINT:PRINT"N
ombre de sprites :";n:PRINT:PRINT"L
ongueur du fichier :";adtab1-depart
+1
790 SAVE n$,b,depart,adtab1-depart+ [3633]
1:CALL &BB06:GOTO 980
800 '                               [117]
810 ' *** IMPLANTATION TABLE        [1613]
820 '                               [117]
830 POKE adimpl,255:POKE adimpl+1,2 [2833]
55
840 adtab=adimpl+2:adtab1=adimpl+3  [3095]
850 g=depart:n=0                    [889]
860 a=PEEK(g):b=PEEK(g+1):IF a=255  [6222]
AND b=255 THEN n=MIN(n,255):POKE ad
tab,n:RETURN
870 n=n+1:msb=INT(g/256):lsb=g-256* [4396]
msb:POKE adtab1,lsb:POKE adtab1+1,m
sb
880 adtab1=adtab1+2:g=g+2+a*b:GOTO  [2528]
860
890 '                               [117]
900 ' *** CHARGEMENT                [648]
910 '                               [117]
920 MODE 1:PEN 2:PRINT"CHARGEMENT F [3437]
ICHIER":PRINT:a$="*.spr":PEN 1:|DIR
,@a$
930 PEN 2:PRINT"Nom du fichier a ch [4733]
arger,sans extension":INPUT"(ENTER
pour abandonner) : ",n$
940 IF n$=""GOTO 980                [1067]
950 n$=LEFT$(n$,8)+".spr":LOAD n$,a [2866]
dimpl:g=depart
960 a=PEEK(g):b=PEEK(g+1):IF a=255  [4746]
AND b=255 THEN adimpl=g:GOTO 980
970 g=g+2+a*b:GOTO 960              [647]
980 '                               [117]
990 ' *** MENU                      [887]
1000 '                              [117]
1010 PEN 1:PAPER 0:MODE 1           [1266]
1020 PEN 3:LOCATE 6,3:PRINT"SPRITER [6898]
     (c) Herve INISAN":LOCATE 6,4:P
EN 2:PRINT"-----------------------
----":PEN 1
1030 LOCATE 6,9:PRINT"1-Chargement  [16850]
ecran":LOCATE 6,11:PRINT"2-Adresse
d'implantation ("depart")":LOCATE 6
,12:PRINT"  Adresse courante :
 "adimpl:LOCATE 6,14:PRINT"3-Sauveg
arde sprites":LOCATE 6,16:PRINT"4-C
hargement sprites"
1040 LOCATE 6,18:PRINT"5-Retour au  [2908]
BASIC"
1050 a$=INKEY$:IF a$<"1" OR a$>"5"G [1819]
OTO 1050
1060 ON VAL(a$)GOTO 260,1070,730,89 [2076]
0,1130
1070 '                              [117]
1080 ' *** ADRESSE D'IMPLANTATION   [2012]
1090 '                              [117]
1100 MODE 1:PEN 2:PRINT"En modifian [11640]
t l'adresse d'implantation,  vous e
ffacez tous les sprites deja crees"
:PRINT:PRINT:PEN 1
1110 PRINT"Adresse d'implantation : [8175]
 &"HEX$(depart)"="depart:PRINT:INPU
T"Nouvelle adresse : ",aa:IF aa=0 G
OTO 980
1120 IF aa<10000 OR aa>42000 GOTO 1 [4531]
100 ELSE depart=aa:adimpl=aa:GOTO 9
80
1130 '                              [117]
1140 ' *** FIN                      [776]
1150 '                              [117]
1160 MODE 1:CALL &BC02:END          [1124]
```

```
1000 '                              [117]
1010 ' *** LM : IMPLANTATION RSX    [2277]
1020 '                              [117]
1030 KEY 138,",":KEY 139,CHR$(13)+" [2017]
data "
1040 DATA 01,09,A0,21,1D,A0,C3,D1,B [3030]
C,11,A0,C3,21,A0,C3,33
1060 DATA A0,54,41,42,4C,C5,53,50,5 [1918]
2,49,54,C5,00,00,00,00
1080 DATA 00,FE,01,C0,DD,6E,00,DD,6 [2737]
6,01,7E,23,32,3B,A0,22
1100 DATA 6E,A0,C9,F3,FE,03,C0,DD,7 [2431]
E,04,FE,00,D0,DD,7E,00
1120 DATA F5,E6,07,5F,F1,CB,3F,CB,3 [2566]
F,CB,3F,D5,11,50,00,CD
1140 DATA B3,A0,D1,7B,E5,11,00,08,C [3405]
D,B3,A0,D1,19,DD,5E,02
1160 DATA 16,C0,19,EB,DD,4E,04,06,0 [3381]
0,CB,21,CB,10,21,10,27
1180 DATA 09,4E,23,46,C5,E1,46,23,4 [2691]
E,23,EB,DD,7E,00,FE,C8
1200 DATA D0,81,3D,FE,C8,D0,DD,7E,0 [3791]
2,FE,50,D0,80,3D,FE,50
1220 DATA D0,C5,06,F5,ED,78,1F,30,F [2653]
B,C1,C5,E5,1A,AE,77,23
1240 DATA 13,10,F9,E1,01,00,08,09,3 [1604]
0,04,01,50,C0,09,C1,0D
1260 DATA 20,E8,C9,06,08,21,00,00,C [3844]
B,2F,30,01,19,CB,23,CB
1280 DATA 12,10,F5,C9               [501]
1290 '                              [117]
1300 MEMORY HIMEM-200:adr=HIMEM+1:s [2077]
um=0
1310 FOR t=0 TO 195:READ a$:a=VAL(" [3217]
&"+a$):POKE adr+t,a:sum=sum+a:NEXT
1320 IF sum<>21890 THEN MODE 1:PRIN [4716]
T"Erreur dans les DATA...":END
1330 '                              [117]
1340 ' *** RELOGEMENT DES RSX       [1539]
1345 '                              [117]
1350 DATA 1,9,4,&1d,9,&11,&c,&21,&f [2995]
,&33,&2d,&3b,&30,&6e,&50,&b3,&59,&b
3
1360 FOR t=1 TO 9:READ a,b          [1552]
1370 c=adr+b:msb=INT(c/256):lsb=c-m [4335]
sb*256:POKE adr+a,lsb:POKE adr+a+1,
msb
1380 NEXT                           [350]
1400 CALL adr                       [311]
```

LECTEURS, n'usez plus vos doigts !

La logithèque d'AM-MAG



Une myriade de messages d'erreur résulte de l'interminable saisie d'un listing prometteur? Votre raison vacille? Nous avons de quoi apaiser vos tourments...

ENFIN PRETE!
La disquette PCW sur laquelle sont regroupés les programmes parus dans nos numéros. **120 F**

3 utilitaires : Vérificateur V.3 - Desfrac - Tailfich.

3 jeux : Othello - La Ville Infernale - Mission Détector.

COMPILATION 140 F

Une disquette regroupe tous les programmes publiés dans nos trois derniers numéros :



n°39



n°40



n°41

Jeux : Reflector, Combat (6128), La roue de la fortune
Utilitaires : Lutins, Fill 464, Tri, Amsaisie, Vérificateurs V.2
Des Dix par Dix à foison **+ un inédit exceptionnel : *MAGE (6128)***

BON DE COMMANDE

Choix

Compilations à déguster

- ☐ **3-4-5** : CPC paint, RSX music, amsmoniteur, Spiderman 2, conjugaisons, le bombardier, dump secteur disc, histogrammes en 3D, récupération fichiers disc, récupération fichiers prot., biorythmes.
- ☐ **6-7-8** : Gloup, taquin, jacquet, A.P.I.D, tennis 3D, rallye 66, pronofoot, musicsoft, statistiques, Amstradascii, crayon opt. 1, crayon opt. 2, casse-briques, géo de France, extensions 464, le compte est bon, Othello.
- ☐ **9-10-11** : M.A.O, gesrevue, Trafalgar, fonctions, Amstermind, labynotaure, musithèques, redéfinition, morts célèbres, verbes irréguliers, nibbler.
- ☐ **12-13-14** : Q-R, loto, X-men, JPG-3D, labybi, gécamob, gest disc, startball, 3D château, extra simon, musithèques, RSX fenêtres, calcul mental, space invaders, Ah! Les vacances, puissance quatre, il pleut bergère, scrolling/déconnect, Amstradian graffitis.
- ☐ **15-16-17** : debug, espace, grille, runner, sphéric, CPC tank, Amsbase, Amspoker, géométrie, oil panic, blockhaus, Mirage IV, warblock, Epson pack, casse-tête, carrés magiques, vérificateurs V.2, monnaies étrangères.
- ☐ **18-19-20** : oil, crypto, pacman, module, rundisc, flipper, dirutil, Amsynth, bowling, G.M.D.B, anagram, fantômes, refusnik, heureduc, culinaire, infernal micro, vérificateurs V.2, gestion de fenêtres.
- ☐ **21-22-23** : spot, astro, pente, combat, Othello, Amsynth, ice strad, boîte à outils, courbes math, curseur flash, arcsin/arccosin, vérificateurs V.2.
- ☐ **24-25-26** : belote, compact, tablatur, analyser, matrices, cer-bert, boursicote man, flight simulator, vérificateurs V.2.
- ☐ **27-28-29** : inversion d'écran, QCM V.2, expert, pluies acides, intro musicale, musique, facture 9, RSX écran, simplification de fraction, supercopter, basic, cauchemar, machaon, mouche, amsaisie, sprites, vérificateurs V.2.
- ☐ **30-31-32** : verlan, utilitaire 208 ko, chasse à l'homme, ku klux klan, dream walker, redstone, gestion souris, caracter, destroy +, suprax, star wars, amsaisie, vérificateurs V.2.
- ☐ **33-34-35** : bouncing creatures, résolution, fenêtrad, traceur de fonctions, blabla technocratique, color, dix par dix, amsaisie, vérificateurs V,2.
- ☐ **36-37-38** : infernal castle, animateur basic, Rock'n Scroll, fantôme, tron, dix par dix, amsaisie, vérificateurs V.2, Spy.
- ☐ **39-40-41** : Lutins en goguette, Reflector, Fill 464, Combat (6128), La roue de la fortune, Tri, dix par dix, amsaisie, vérificateurs V.2, Mage (6128).

☐ Je commande la disquette pour PCW : 120 F

Règlement par ☐ Chèque bancaire ☐ Chèque postal ☐ Mandat

Nom : ____________ Prénom ____________

Adresse : ____________

A retourner à Laser Presse AM MAG, Service Diffusion
5/7, rue de l'Amiral Courbet 94160 SAINT-MANDE
Egalement utilisable, le bon de commande des anciens numéros cassettes, disquettes.

Prix dégressif

1 disquette : 140 f

2 disquettes : 120 f l'ex. soit 240 f

3 disquettes : 100 f l'ex. soit 300 f

4 disquettes : 90 f l'ex. soit 360 f
+ 90 f pour chaque disquette supplémentaire.

Disquettes à 90 F,
pour une commande à partir de 4 exemplaires.

AUTEURS

Nous attendons vos réalisations! Vous programmez et vous souhaitez faire profiter nos lecteurs de vos productions originales. Faites nous parvenir ce bon dûment rempli.

Je soussigné ____________

Adresse : ____________

Certifie sur l'honneur être l'auteur de ce programme et autorise AM MAG à le publier sous formes diverses (revues, sup. disquettes)

Date et signature obligatoires ____________

Tous nos listings parus sont rémunérés.


ABONNEZ-VOUS A AM MAG
TOUT POUR VOTRE AMSTRAD
Chaque mois, le magazine qui vous donne la meilleure couverture de l'actualité AMSTRAD
12 NUMEROS + 4 HORS-SERIE
275 F au lieu de 364 F(*)
* : Prix de vente au numéro : 12 numéros à 22 F, + 4 H.S. à 25 F
Le tarif de l'abonnement 88 est maintenu malgré l'augmentation du prix de vente au numéro.
89 F d'ECONOMIE + CADEAU
Au choix : 2 numéros HORS-SERIE parus en 88
(Cocher dans le bon d'abonnement les numéros choisis)
ou : 1 K7 de jeu
3 super hits de Hewson
URIDIUM
En 86, Jeu de l'année
BATTLE VALLEY
Capturer les terroristes et détruire les missiles
OCEAN CONQUEROR
Aventures sous-marines
(Choix respecté dans la limite des stocks).
N° 5 - SPECIAL LISTINGS
N° 6 - SPECIAL LISTINGS
N° 7 - SPECIAL LISTINGS
N° 8 - GUIDE DES LOGICIELS
N° 9 - AMSTRAD DE A à Z
GONFLEZ VOTRE CPC: LE PLEIN DE COMMANDES NOUVELLES !
JEUX
Thunderblade
Fernandez Must Die
Trivial Pursuit
Dossier Kha
Tiger Road
Pacmania
etc.
AM PRO
GEREZ VOTRE CAVE A VINS
SOIGNEZ VOTRE ORTHOGRAPHE
accessoires et compilations-jeux
Emmanuelle, Cybernoid II, Purple Saturn Day, Titan, Puffy's Saga, Crazy Cars II, Artura, A 320, Galactic Conqueror, 944 Turbo Cup.
AMSTRAD SPECIAL LISTINGS
HORS-SERIE N° 6
les meilleurs inédits sur CPC et PCW
GUIDE DES JEUX !
PLUS DE 500 LOGICIELS
AMSTRAD de A à Z
Abonnement à AM MAG
☐ Oui, je m'abonne à AM MAG pour 12 numéros + 4 H.S. au prix de 275 F (au lieu de 364 F)
☐ Europe : 375 F ☐ Armail : 435 F
☐ Je choisis en cadeau 2 numéros hors-série : ☐ n° 5, ☐ n° 6, ☐ n° 7, ☐ n° 8, ☐ n° 9
ou 1 K7 de jeu : ☐ Battle Valley, ☐ Uridium, ☐ Ocean Conqueror
Je règle par : ☐ chèque bancaire, ☐ chèque postal, ☐ mandat.
à adresser à :
AM MAG
Service Diffusion
5/7, rue de l'Amiral Courbet
94160 Saint-Mandé.
Nom : Prénom
Adresse :
Ville Code postal
ATTENTION ! Offre limitée jusqu'au 01/04/1989